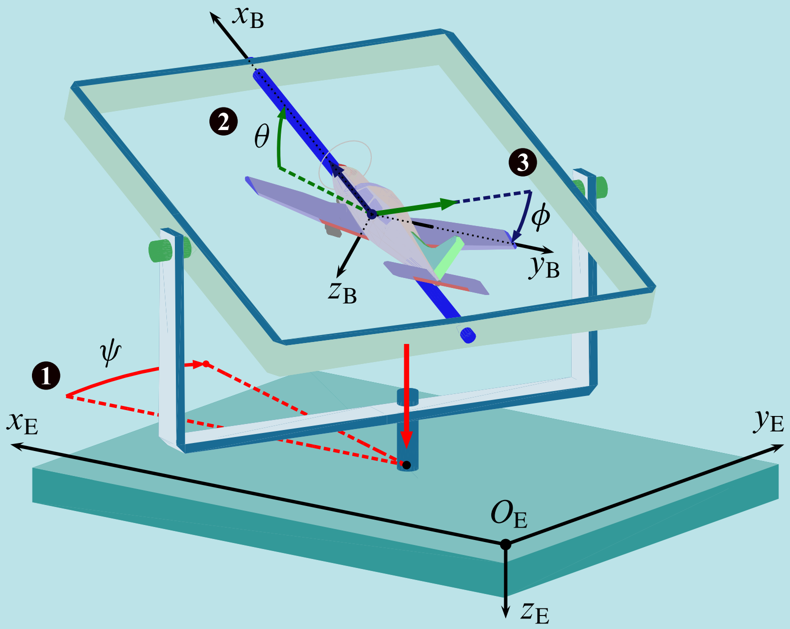

Flight Dynamics with Matlab/Simulink

Elaborato di Dinamica e simulazione di volo

Part II

Antonio Carotenuto

A.A 2023-2024

Indice

Elenco delle figure

Capitolo 1

Control Surfaces

1.1 Introduction

The aircraft equations of motion seen so far have been derived under the of the rigid body assumption, which implies a fixed configuration. The possibility of deflecting mobile surfaces suggests that this hypothesis is not strictly verified. However, the shape variations resulting from the rotations of said surfaces do not determine, in relation to the large dimensions of the complete aircraft, a significant variation in mass distribution. As a consequence, it is possible to apply the rigid body hypothesis to our problem while still allowing non-zero angular excursions of the control surfaces. It can be interesting to study the dynamics of control surfaces because knowledge of the loads acting on them is fundamental, and in some cases, the excursions of the controls can themselves represent unknowns, such as during motion with free controls. A control mechanism can be schematized as a rigid articulation constrained to rotate around an axis fixed to the aircraft, which we will call the *hinge axis*. It is useful to introduce a reference system fixed to the control surface, which we denote by *CS* (Control Surface) and is defined as follows:

- $x_{CS} = c, y_{CS} = t, z_{CS} = n$;
- The origin is identified by the intersection point between the longitudinal plane of symmetry and the hinge axis;
- the first axis is the hinge axis;
- the second axis passes through the center of gravity;
- the n-axis is such that it completes a left-handed triad;

The motion of the generic control surface is governed by the rotational equilibrium equation, projected along the hinge axis:

$$\left(\dot{\mathcal{K}}_{\mathrm{r,CS}}\right)_c + (\Omega_{\mathrm{CS}})_t \left(\mathcal{K}_{\mathrm{r,CS}}\right)_n - (\Omega_{\mathrm{CS}})_n \left(\mathcal{K}_{\mathrm{r,CS}}\right)_t + m_{\mathrm{CS}} \left(a_C\right)_n e_{\mathrm{CS}} = \mathcal{M}_c \tag{1.1}$$

It is possible to simplify the equation by making the following assumptions:

- the ct-plane is assumed as the plane of symmetry of the solid, consequently the products of inertia with respect to the pairs of axes of the reference frame are null.
- The control surfaces are like a lamina contained in the plane of the c and t unit vectors; at this point, we can express one moment of inertia as a function of the other two, in our case we obtain: $I_n = I_c + I_t$;
- since the deflections are small, we can consider that the ct-plane of symmetry is always in one of the coordinate planes of the body reference frame;

It is important to note that, although the control surface is fixed to the aircraft, it can rotate independently, and therefore it is necessary to introduce an angular velocity of the CS frame with respect to the body axes of the aircraft ω_{CS}, and an angular velocity of the CS frame with respect to the inertial reference frame Ω_{CS}. The following relationship holds:

$$\Omega_{CS} = \Omega_B + \omega_{CS} \tag{1.2}$$

In this case, the forcing term to consider is the hinge moment, which we model as the sum of three contributions.

- $\mathbf{H}_{\mathbf{A_{CS}}}$ of aerodynamic nature
- $\mathbf{H}_{\mathbf{g_{CS}}}$ due to the weight force applied at the hinge point
- $\mathbf{H}_{\mathbf{C_{CS}}}$ due to the pilot's action on the controls

In particular, $\mathbf{H}_{\mathbf{g_{CS}}} = m_{CS} g_n e_{CS}$ where e_{CS} is called *eccentricity* and is the distance between the center of gravity of the control surface and the hinge point. If the hinge point and the center of gravity of the control surface coincide, the eccentricity will be null, and the aircraft will be said to be *statically balanced*. In light of the simplifying assumptions and observations made, the eq that governs the motion is rewritten as:

$\mathbf{\Omega_{CS}}$ in the CS reference frame, we obtain:

$$I_c \ddot{\delta}_{CS} - (\dot{p}+qr) I_c \sin\Lambda_C + (\dot{q}-pr) I_c \cos\Lambda_C + m_{CS}(a_{G_{Z_B}} - g_{Z_B}) e_{CS} = \mathcal{H}_{A,CS} + \mathcal{H}_{C,CS} \tag{1.3}$$

The first term is the classic term present in a second-order pendulum-type dynamic, the second and third are inertial couples due to the coupling of the control surface dynamics with the aircraft motion. The fourth is due to a possible eccentricity and is a hinge moment; if we perform a *static balancing*, this term is null. On the right-hand side are the hinge moments of aerodynamic and control nature. Using this notation:

$$\begin{cases} I_{cx_B} = m_{CS} e_{CS} y_{B,C} - I_c \sin\Lambda_c \\ I_{cy_B} = m_{CS} e_{CS} x_{B,C} - I_c \cos\Lambda_c \end{cases} \tag{1.4}$$

The previous equation becomes:

$$I_c \ddot{\delta}_{CS} + (\dot{p}+qr) I_{cx_B} - (\dot{q}-pr) I_{cy_B} + m_{CS}(a_{G_{Z_B}} - g_{Z_B}) e_{CS} = \mathcal{H}_{A,CS} + \mathcal{H}_{C,CS} \tag{1.5}$$

1.2 Longitudinal Control

The preceding equations are valid for any control surface and can be specified to study the dynamics of the elevator following a simple change of symbology. For the elevator, it results: $I_{h_e x_B} = 0$. If the eccentricity is null ($e_e = 0$), the equation simplifies and becomes:

$$I_{h_e}\ddot{\delta}_e - (\dot{q} - pr)I_{h_e y_B} = \mathcal{H}_{A,e} + \mathcal{H}_{C,e} \tag{1.6}$$

Dynamic coupling terms are not desired; it is noted from the equations that an angular acceleration induces an aerodynamic hinge moment (with free controls) or a greater control effort (with fixed controls). Similarly for linear accelerations. Static balancing can be achieved by nullifying the eccentricity, i.e., by positioning the center of gravity on the hinge axis. For the elevator, dynamic balancing cannot be achieved; indeed, $I_{h_e y_B}$ can only be nullified for large $Lambda_C$ angles, which do not have physical meaning. If the motion develops with free controls, then the control hinge moment is null, and the equation is modified accordingly. In the case of free controls, we observe that the motion of the elevator depends on aerodynamic actions and is dependent on the aircraft's dynamics. This, in turn, is influenced by the excursion of the mobile surface. A system of equations to be solved is thus obtained, which is similar to that seen in the previous chapter, but the elevator deflection can no longer be considered a control law but an unknown. An aerodynamic model for the hinge moment needs to be implemented; under the hypothesis of small angles and low Strouhal number, the following linearized model is plausible:

$$C_{\mathcal{H}_e} = C_{\mathcal{H}_0} + C_{\mathcal{H}_\alpha}\alpha_H + C_{\mathcal{H}_{\delta_e}}\delta_e + C_{\mathcal{H}_{\delta_s}}\delta_s + C_{\mathcal{H}_{\delta_t}}\delta_t + \frac{\bar{c}_e}{2V}\left[C_{\mathcal{H}_{\dot{\alpha}}}\dot{\alpha_H} + C_{\mathcal{H}q}q + C_{\mathcal{H}_{\dot{\delta}_e}}\dot{\delta}_e\right] \tag{1.7}$$

$$\alpha_H = \left(1 - \frac{d\epsilon}{d\alpha}\right)\alpha_B - \epsilon_0 + \delta_s + \mu_x \Rightarrow \dot{\alpha}_H = \left(1 - \frac{d\epsilon}{d\alpha}\right)\dot{\alpha} \tag{1.8}$$

The system of first-order differential equations to be solved becomes:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \\ 0 & M_{32} & 1 & 0 & 0 & 0 & 0 & M_{38} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ M_{71} & M_{72} & M_{73} & 0 & 0 & M_{76} & 1 & M_{78} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{Bmatrix} \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{Bmatrix} \tag{1.9}$$

Where f_1, f_2, f_4, f_5, f_6 are unchanged and equal to the equations for fixed controls with x_8 instead of u_2 (the elevator deflection is no longer a control variable but an unknown). The matrix on the LHS is a *mass matrix*; the library function *ode45* allows us to solve problems also in the form:

$$[\boldsymbol{M}(t, \boldsymbol{x})\,\{\dot{\boldsymbol{x}}\} = \{f(t, \boldsymbol{x})\}] \tag{1.10}$$

the matrix M can be defined by setting up an appropriate function whose pointer will be made known to *ode45* through the *odeset* function.

1.3 Solution of Equations of Motion with Free Controls

A calculation code has been developed to implement the equations previously proposed for the study of longitudinal symmetric motion in the case of free controls. For completeness, the form of the system type that was solved below is reported.

$$\left\{ \dot{x}(t) \right\} = [M(x,t)]^{-1} \left\{ f(x,u,t) \right\} \tag{1.11}$$

For a certain flight time $t_1 = 1s$, the controls are kept fixed, so the equations seen in the previous chapter apply. From t_1 up to t_f, the controls are left free, so the equations just seen apply. Below is the calculation code and the outputs consistent with the physics of the problem. It is also observed how the results vary with the position of the center of gravity. The static pitch stability $\frac{dC_M}{d\alpha}$=$C_{M\alpha}$ depends on this position:

$$C_{M\alpha} = -C_{L\alpha}(\overline{x_N} - \overline{x_G}); \tag{1.12}$$

Calculations were performed considering the following 4 center of gravity positions, with $\overline{x_N}$ =0.45

- $\overline{x_G}$=0.27$\Rightarrow C_{M_\alpha} < 0$
- $\overline{x_G}$=0.32$\Rightarrow C_{M_\alpha} < 0$
- $\overline{x_G}$=0.37$\Rightarrow C_{M_\alpha} < 0$
- $\overline{x_G}$=0.43$\Rightarrow C_{M_\alpha} < 0$

In the case where the center of gravity coincides with the neutral point, i.e., the condition for which the stability line loses slope until it becomes zero slope, the aircraft is said to have neutral stability. In the case of an aircraft with free controls, the static pitch stability index is certainly lower because the position of the neutral point changes, which will be slightly lower than $\overline{x_N}$ =0.45. Therefore, with free controls, the aircraft loses stability for less aft center of gravity positions.

Listing 1.1

```
global g...                         %Accelerazione di gravita'
       zEG_0 V0 q0 gamma0...        %Condizioni iniziali
       rho0 ...                     %Densit  dell'aria all'altitudine h = (-
   zEG_0)
       myAC    ...                  %Oggetto 'Velivolo'
       delta_e...
       delta_s...
       delta_T ...
       delta_tab
%% Populate aircraft data
aircraftDataFileName = 'DSV_Aircraft_data.txt';

%% Aircraft object
```

```
myAC = DSVAircraft(aircraftDataFileName);

if (myAC.err == -1)
    disp('... terminating.');
else
    disp(['File ', aircraftDataFileName, ' letto correttamente.']);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% MODIFICO QUA %%

%% VOGLIO FARE STUDIO PARAMETRICO, VEDO COSA ACCADE AL VARIARE DELLA POS
%% DEL BARICENTRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    %  Allocazione in memoria delle matrici di raccolta dati
    trim_matrix = zeros(5,4);

    X_CG_vector = [myAC.Xcg_adim-0.05,myAC.Xcg_adim,myAC.Xcg_adim
   +0.05,0.42];
    for i = 1:4

        myAC.Xcg_adim = X_CG_vector(i);
        myAC.Cm_alpha = -myAC.CL_alpha*(myAC.Xn_adim - myAC.Xcg_adim);


    %% Condizioni iniziali
    xEG_0=0;
    zEG_0 = -4000; % =quota (m)
    V0 = 257; % velocita' di volo
    q0 = 0; % velocita ' angolare di beccheggio (rad/s)
    gamma0 = convang(0, 'deg', 'rad'); % angolo di salita (rad)

    %% Densita ' con modello ISA
    [air_Temp0, sound_speed0, air_pressure0, rho0] = atmosisa(-zEG_0);

    %% Accelerazione di gravita '
    g = 9.81; % (m/s  2)

    %% Tentativo iniziale
    x0 = [
               0;  %alpha0
               0;  %deltae0
               0;  %deltas0
               0.5 % delta_T_0
           ];
    %% Minimo della funzione di costo
    % Aeq, in Aeq*x=beq linear constraint
    Aeq = zeros (4, 4);
    ceq = zeros (4, 1);
    Aeq(3, 3) = 1;
    delta_s_0 = convang(-0.0, 'deg', 'rad');
    ceq(3, 1) = delta_s_0; %tiene deleta_s fissato
    % bounds
```

```
lb = [convang(-15, 'deg', 'rad'), ... % minimo alpha
        convang(-20, 'deg', 'rad'), ... % minima deflessione dell '
equilibratore
        convang(-5, 'deg', 'rad'), ... % minima incidenza dello
stabilizzatore
        0.2]; ... % minima manetta
    ub = [convang(15, 'deg', 'rad'), ... % massimo alpha
        convang(13, 'deg', 'rad'), ... % massima deflessione dell '
equilibratore
        convang(2, 'deg', 'rad'), ... % massima incidenza dello
stabilizzatore
        1.0]; %massima manetta
options = optimset( ...
    'tolfun', 1e-9, ... %treshold
    'Algorithm', 'interior-point' ... % algor. type
);
[x, fval] = ...
    fmincon(@costLongEquilibriumStaticStickFixed, ...
    x0, ...
    [], ... %A,A*x<=b
    [], ... %b
    Aeq, ... % Aeq , Aeq*x=beq
    ceq, ... % beq
    lb, ub, ...
    @myNonLinearConstraint, ...
    options);

alpha0_rad = x(1);
alpha_0_deg = convang(alpha0_rad, 'rad', 'deg');

theta0_rad = alpha0_rad - myAC.mu_x + gamma0;
theta_0_deg = convang(theta0_rad, 'rad', 'deg');

delta_e0_rad = x(2);
delta_e_0_deg = convang(delta_e0_rad, 'rad', 'deg');

delta_s0_rad = x(3);
delta_s_0_deg = convang(x(3), 'rad', 'deg');

delta_T0 = x(4);

t_1=1; %istante in cui si lasciano i comandi
t_fin=30; %istante finale
state0 = [V0,alpha0_rad,q0,xEG_0,zEG_0,theta0_rad]; %vettore di stato
 iniziale
%della dinamica a comandi liberi, ha come componenti gli elementi
della
%condizione di trim

% Assegnazione delle leggi temporali dei comandi di volo
delta_e = @(t) interp1([0,  t_1],...
                        [delta_e0_rad, delta_e0_rad],...
                        t,'linear');
delta_s = @(t) interp1([0,t_fin],[delta_s0_rad,delta_s0_rad],t,'
linear');%cost
delta_T = @(t) interp1([0,t_fin],[delta_T0,delta_T0],t,'linear');%
```

```
    cost
    delta_tab = @(t) 0*t; %fissato a 0
%disp('')
%disp('Condizione di trim:')
%disp(['Velocit  V_0= ',num2str(V0), 'm/s'])
%disp(['Angolo d%attacco alpha_0=' ,num2str(alpha_0_deg),'deg'])
%disp(['Elevatore delta_e_0= ',num2str(delta_e_0_deg),'deg'])
%disp(['Stabilizzatore delta_s_0= ',num2str(delta_s_0_deg),'deg'])
%disp(['Manetta delta_T_0= ',num2str(delta_T0)])
%disp(['Angolo del Tab=0',0])

%% PASSO2 Integrazione d e l l istante  t0=0s a l l istante  t1=1s comandi bloccati
%Integrazione a comandi bloccati
[T1,Y1]=ode45(@eqLongDynamicStickFixed_,[0 t_1],state0);
%estrazione delle variabili
V1=Y1(:,1);
alpha1=Y1(:,2);
q1=Y1(:,3);
xeg1=Y1(:,4);
zeg1=Y1(:,5);
gamma1=Y1(:,6);
theta1=gamma1+alpha1-myAC.mu_x;
% Tutti i vettori ottenuti sono le condizioni di trim protratte fino all' istante 1
% prevedibilmente sono costanti
%% PASSO3 Risoluzione del sistema di equazioni 7.42, d a l l istante  t1=1s a
% tf=30s per il caso di comandi liberi

delta_e0_dot_rad = convangvel(0,'deg/s','rad/s');
state_2 = [V1(end),alpha1(end),q1(end),xeg1(end),zeg1(end),theta1(end),...
             delta_e0_dot_rad,delta_e(T1(end))];
[T2,Y2] = ode45(@eqLongDynamicStickFree_,[t_1 t_fin],state_2);
%Estrazione grandezze di interesse
V2=Y2(:,1);
alpha2=Y2(:,2);
q2=Y2(:,3);
xeg2=Y2(:,4);
zeg2=Y2(:,5);
theta2=Y2(:,6);
delta_e2_dot_rad=Y2(:,7);
delta_e2=Y2(:,8);
%Composizione dei vettori di stato di tutta la manovra:(vettori colonna)
T=[T1; T2]; %[s]
V=[V1; V2]; %m/s
alpha=[alpha1; alpha2]; %rad
q=[q1; q2]; %rad/s
xeg=[xeg1; xeg2]; %m
zeg=[zeg1; zeg2]; %m
theta=[theta1; theta2]; %rad
gamma=theta-alpha+myAC.mu_x; %[rad]
dot_delta_e=[zeros(length(T1),1); delta_e2_dot_rad];  %rad/s
delta_e_=[delta_e0_rad*ones(length(T1),1); delta_e2]; %rad

```

```
%% modifico qua per costruire matrice m_state
m_state=[T,V,alpha,q,xeg,zeg,theta,gamma,dot_delta_e,delta_e_];

%% creato matrice di stato da t=0 a t=tfin


alpha_dot=gradient(alpha,T); %rad/s
v_dot=gradient(V,T);
q_dot=gradient(q,T); %[rad/s]
q_dot_deg_s=convangvel(q_dot,'rad/s','deg/s'); %deg/s
gamma_dot=gradient(gamma,T); %rad/s
fxa = -sin(gamma)-v_dot./g;
fza    = cos(gamma)+(gamma_dot.*V)./g;
% Angolo d'attacco del piano di coda orizzontale
    alpha_H_rad = (1 - myAC.DepsDalpha)*(alpha-myAC.mu_x) - myAC.eps_0
   +...
                    delta_s(T) + myAC.mu_x; %rad
    alpha_H_rad_dot = (1 - myAC.DepsDalpha)*(alpha_dot); %rad/s

% Coefficiente momento di Ceniera
    v_Ch_e = myAC.Ch_e_0 + myAC.Ch_e_alpha*alpha_H_rad +...
             myAC.Ch_e_delta_e*delta_e_+...
             myAC.Ch_e_delta_s*delta_s(T)+...
             myAC.Ch_e_delta_tab*delta_tab(T)+...
             (myAC.mac_e/(2*V))...
    *(myAC.Ch_e_alpha_dot*alpha_H_rad_dot+...
                                  myAC.Ch_e_q*q+...
                                  myAC.Ch_e_delta_e_dot*dot_delta_e);
    % Momento aerodinamico di cerniera
    v_H_Ae = 0.5*density(-zeg).*V.^2*myAC.S_e*myAC.mac_e.*v_Ch_e;
```

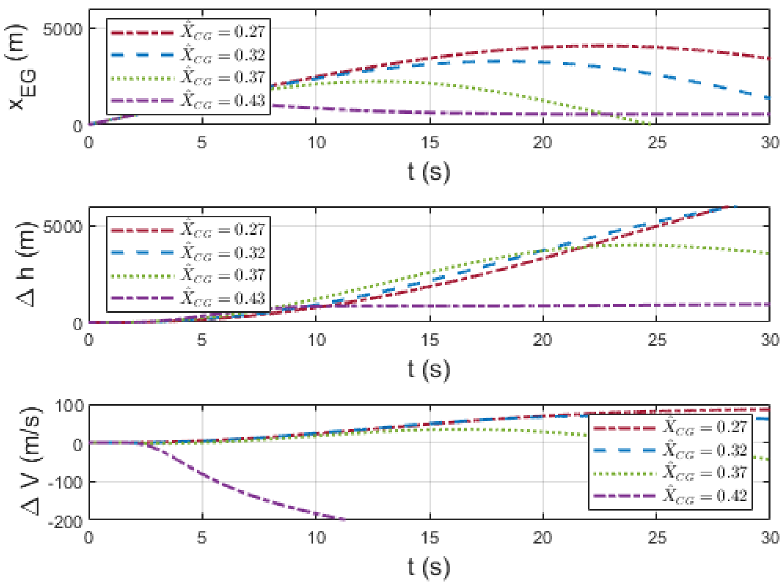

Figura 1.1 Time histories of state variables for fixed controls until time $t_1 = 1$s and free controls in the subsequent time instants. The quantities $\Delta(*)$represent deviations from initial values.

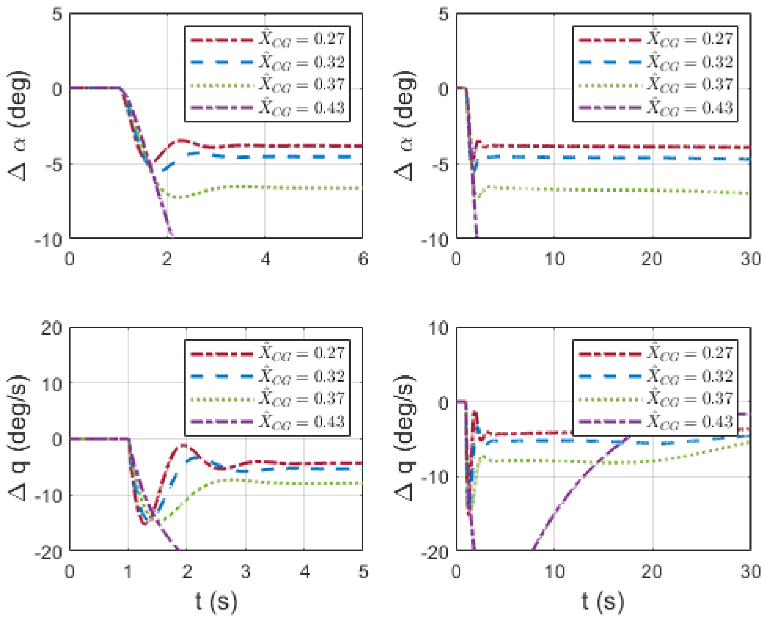

Figura 1.2 Time histories of state variables for fixed controls until time t_1 = 1s and free controls in the subsequent time instants. The quantities $\Delta(*)$represent deviations from initial values

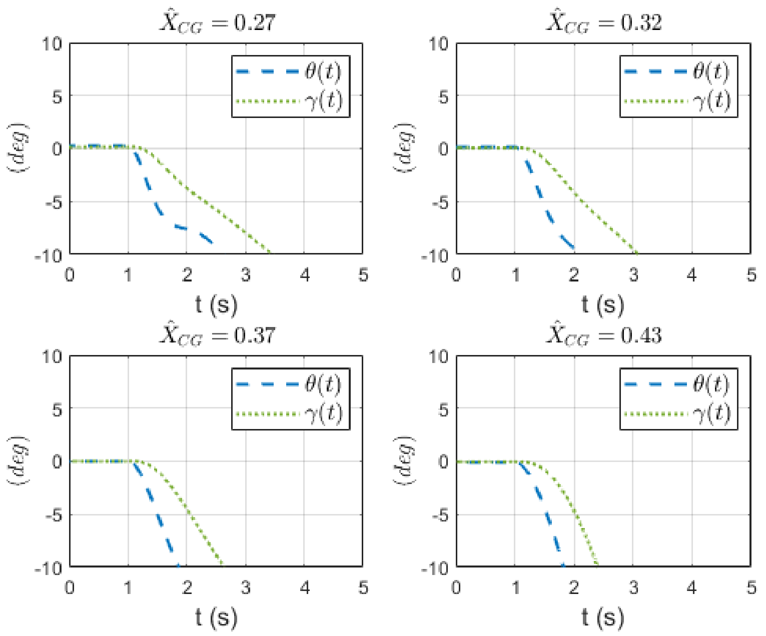

Figura 1.3 Time histories of state variables for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

It is interesting to observe that in the case of a very aft center of gravity x_G=0.43, we see that the aircraft responds to the pilot's release of controls violently, and the response dampens more slowly compared to the other 3 cases. High variations in angle of attack and pitch angular velocity are observed. Very violent and rapid oscillations of the elevator are observed. This results in very high load factors being reached. In particular, the normal load factor is greater than 5, and not even an experienced pilot equipped with an anti-g suit can withstand such high values. There would also be structural problems. It is observed that the further aft the center of gravity, the faster the oscillations around

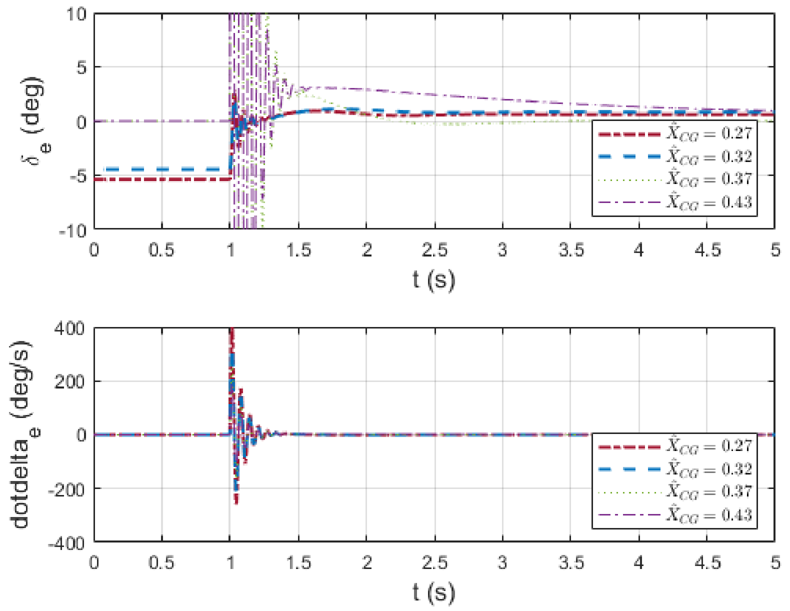

Figura 1.4 Time histories of the variation of δ_e, its rate of change $\dot{\delta}_e$, for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

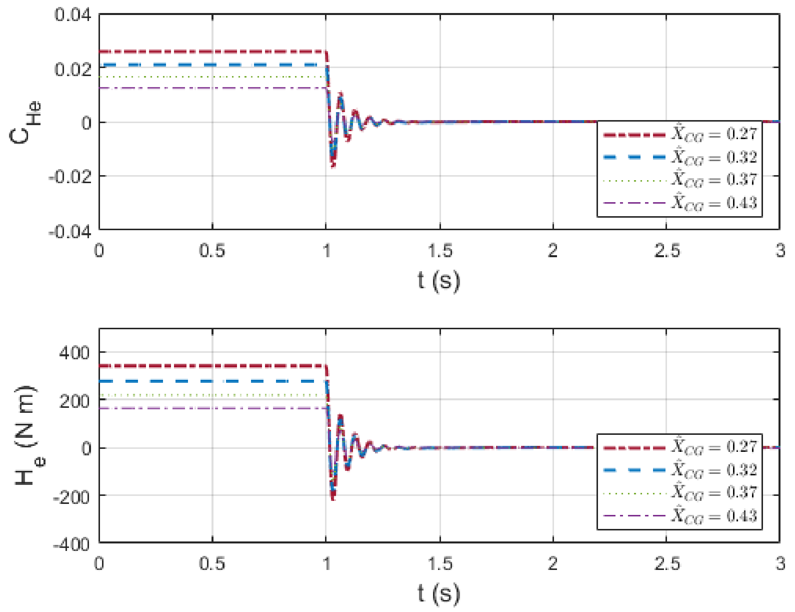

Figura 1.5 Time histories of the variation of the hinge moment coefficient C_{He} and the hinge moment $\mathbf{H_e}$for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

the final equilibrium values of angle of attack, flight path angle, elevator deflection angle, and load factors are damped. Furthermore, the total variations of these quantities are lower.

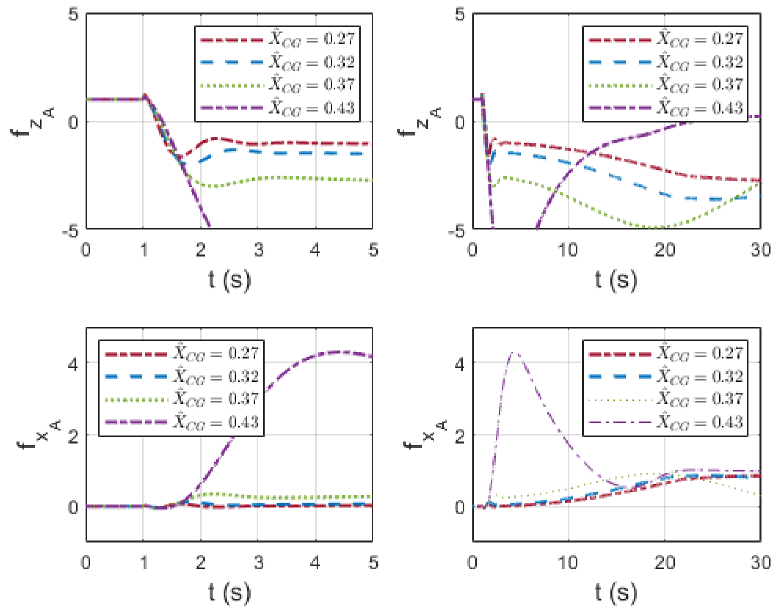

Figura 1.6 Time histories of the load factor along the z_A and x_A axes for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

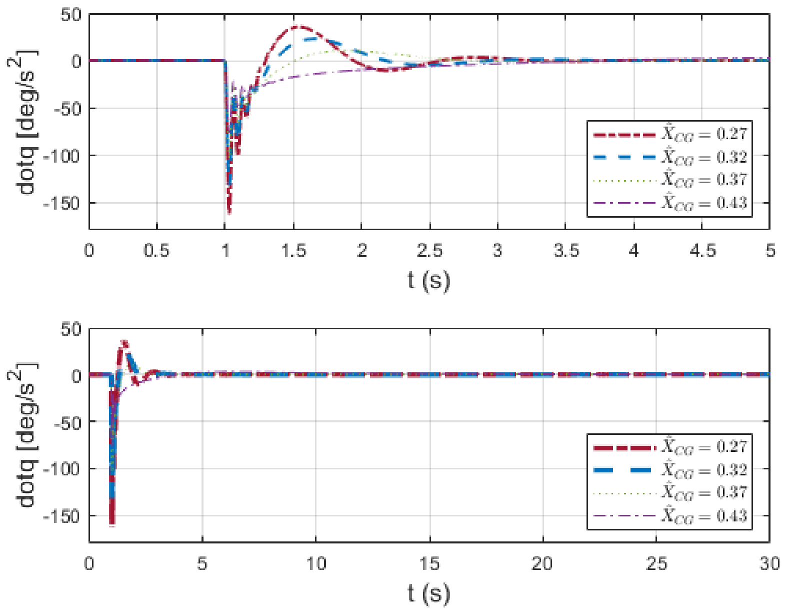

Figura 1.7 Time histories of the angular acceleration $\dot{q}$ for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

1.4 Introduction to Trim Tab

The objective of this exercise is to find the deflection δ_t necessary to obtain an asymptotic angle δ_e equal to δ_{e_0}, i.e., that obtained by imposing trim conditions.

In asymptotic conditions, only the aerodynamic hinge moment and the moment due to the eccentricity of the center of gravity with respect to the hinge axis act on the elevator. Therefore, assuming that this condition coincides with the initial trim condition, the rotational equilibrium equation becomes:

$$\begin{aligned} m_e\left(a_{G_{z_B,0}} - g_{z_B,0}\right) e_e = \mathcal{H}_{A,e} = \\ = \bar{q}_\infty S_e \bar{c}_e \left[C_{\mathcal{H}_0} + C_{\mathcal{H}_\alpha}\alpha_H + C_{\mathcal{H}_{\delta_e}}\delta_{e,0} + C_{\mathcal{H}_{\delta_s}}\delta_s + C_{\mathcal{H}_{\delta_t}}\delta_t^*\right] \end{aligned} \tag{1.13}$$

This equation must therefore be solved to derive δ_t^*:

$$C_{\mathcal{H}_{\delta_t}}\delta_t^* = \frac{m_e\left(a_{G_{z_B,0}} - g_{z_B,0}\right) e_e}{q_\infty S_e c_e} - \left[C_{\mathcal{H}_0} + C_{\mathcal{H}_\alpha}\alpha_H + C_{\mathcal{H}_{\delta_e}}\delta_{e,0} + C_{\mathcal{H}_{\delta_s}}\delta_s\right] \tag{1.14}$$

In our case, the deflection value of the *trim tab* that nullifies the hinge moment is $\delta_t^* = 2.19°$ Below is the code, which is the same as the previous code, but a control law is inserted that brings the tab value from 0 to the value δ_t^* in one second.

Listing 1.2

```
% Calcolo Trim Tab
theta_trim = gamma0 + alpha0_rad - myAC.mu_x;
alpha_body_0 = alpha0_rad-myAC.mu_x;                   % alpha body
alpha_H_0 = (1-myAC.DepsDalpha)*(alpha_body_0)...    % alpha H
             -myAC.eps_0 + delta_s0_rad + myAC.mu_x;
CH_A_e_delta_tab_segnato = ((myAC.mass_e*myAC.ec_adim*myAC.mac_e)...
         *g*cos(theta_trim))/(0.5*rho0*V0^2*myAC.S_e*myAC.mac_e)...
         - (myAC.Ch_e_0 + myAC.Ch_e_alpha*alpha_H_0 ...
         + myAC.Ch_e_delta_e*delta_e0_rad ...
         + myAC.Ch_e_delta_s*delta_s0_rad);
delta_tab_segnato = CH_A_e_delta_tab_segnato/myAC.Ch_e_delta_tab;
%% modifico qua faccio due leggi della manetta
for i=1:3
    t_1=1;
    t_2=10;
    t_fin=150;

    if i==1

    t_3=15;
    t_4=20;
    delta_tab = @(t) interp1([0, t_1, t_2, t_3, t_4, t_fin], ...
        [0, delta_tab_segnato, delta_tab_segnato, delta_tab_segnato*3/5,
   delta_tab_segnato, delta_tab_segnato] ...
        , t,'linear');
    elseif i==2

    t_3=20; %25
    t_4=30; %40
    delta_tab = @(t) interp1([0, t_1, t_2, t_3, t_4, t_fin], ...
        [0, delta_tab_segnato, delta_tab_segnato, delta_tab_segnato*3/5,
   delta_tab_segnato, delta_tab_segnato] ...
        , t,'linear');
    elseif i==3
    t_3=15; %t_3=20
    t_4=20;
    delta_tab = @(t) interp1([0, t_1, t_2, t_3, t_4, t_fin], ...
        [0, delta_tab_segnato, delta_tab_segnato, delta_tab_segnato*1/5,
```

```
    delta_tab_segnato, delta_tab_segnato] ...
        , t,'linear');
    end
```

It is interesting to visualize the outputs obtained between 1 s and 10 s, considering the downward deflection of the trim tab that ensures proceeding in trim conditions. We expect, respecting the physics of the system, that there are no variations in angular velocities and load factors. After 10 s, we assume that the pilot wants to bring the aircraft to a slightly higher altitude without intervening on the elevator or throttle. This can be done by modifying the excursion of the trim tab, deflecting it first downwards and then upwards according to linear trends. The results for three different time laws are reported.

We observe that the altitude variation increases if we increase either the time during

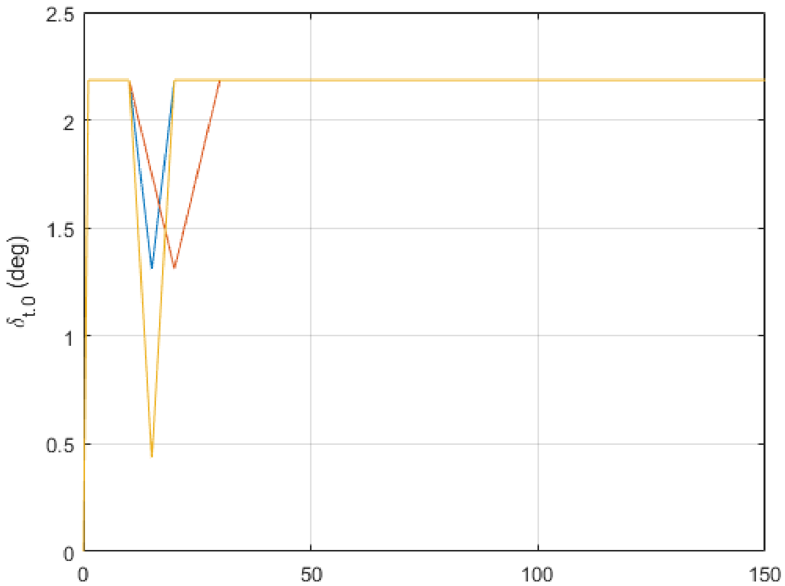

Figura 1.8 Time variation of the quantity Δ_{tab} with linear trend in the first second of observation, constant up to 10 seconds then linearly decreasing and subsequently linearly increasing trend

which the trim tab deflection angle is varied, or by increasing the maximum excursion of the trim tab. It is also observed that proceeding in the first way achieves lower load factors compared to the second case.

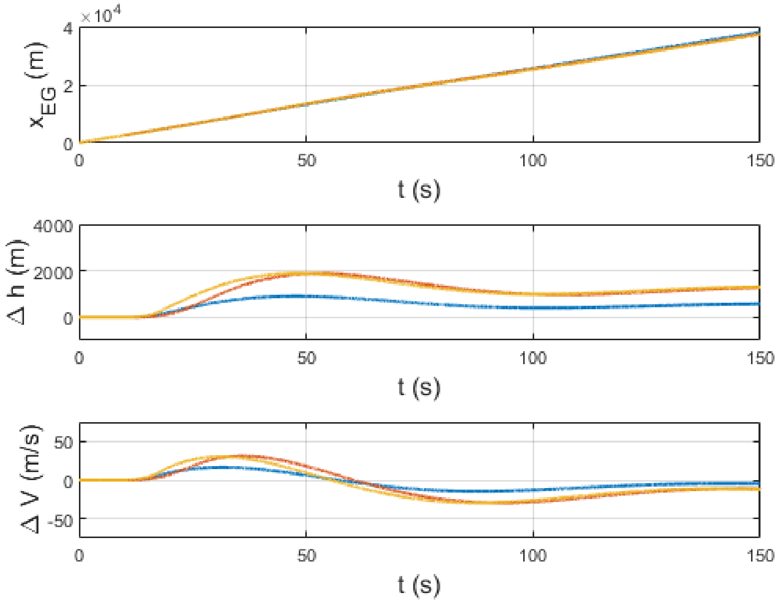

Figura 1.9 Time histories of state variables for fixed controls until time t_1 = 1s and free controls in the subsequent time instants. The quantities $\Delta(*)$represent deviations from initial values.

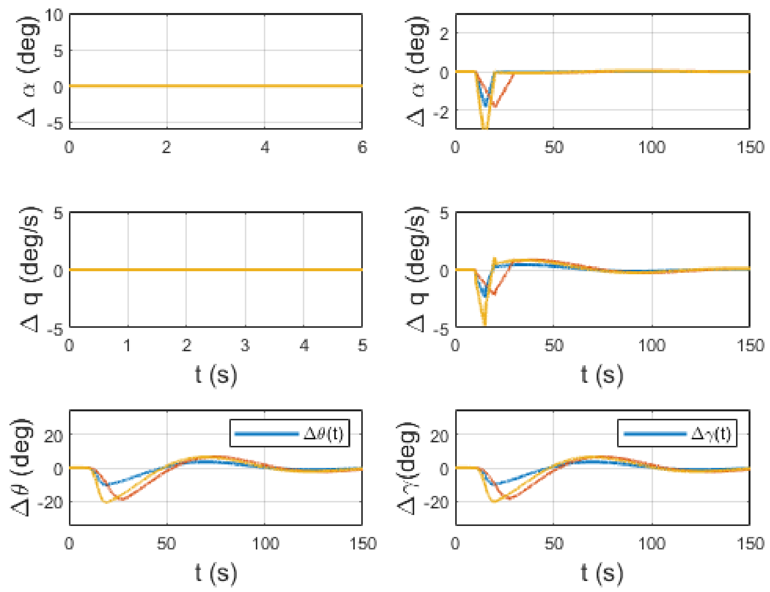

Figura 1.10 Time histories of state variables for fixed controls until time t_1 = 1s and free controls in the subsequent time instants. The quantities $\Delta(*)$represent deviations from initial values

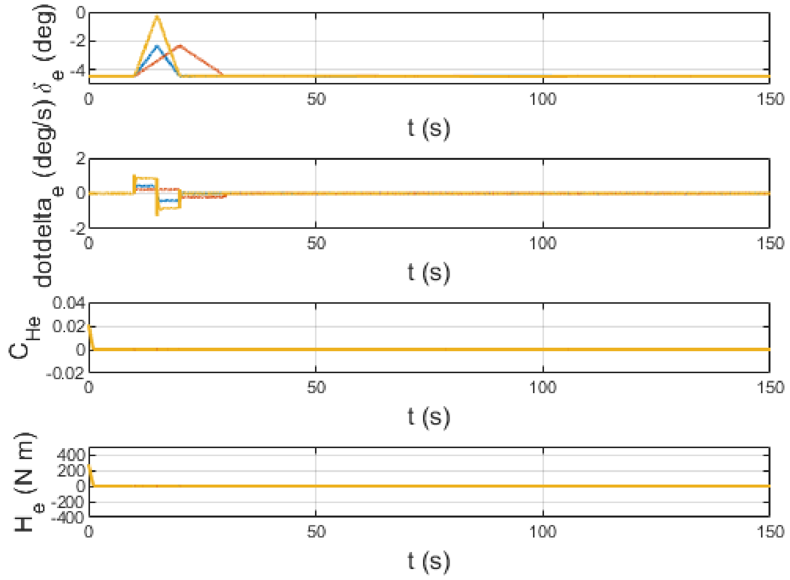

Figura 1.11 Time histories of the variation of δ_e, its rate of change $\dot{\delta}_e$, of the hinge moment coefficient C_{He} and of the hinge moment $\mathbf{H_e}$ for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

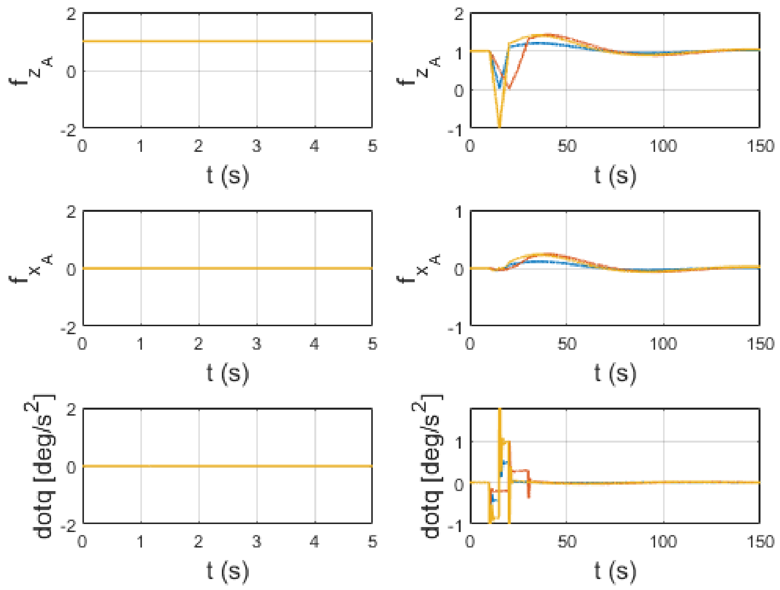

Figura 1.12 Time histories of the load factor along the z_A and x_A axes for fixed controls until time t_1 = 1s and free controls in the subsequent time instants.

Capitolo 2

Small Perturbations in Aircraft Motion

2.1 Introduction

We have observed in the previous chapters that the study of aircraft motion cannot disregard considerations on the complexity of the aircraft system and that it is necessary to take into account the subsystems that compose it, such as propulsion systems and control systems like mobile surfaces. We have observed that the motion of the aircraft, under the hypotheses of symmetry with respect to the longitudinal plane, is governed by the following system of equations:

$$\begin{cases} m(\dot{u} + qv - rv) = X_G + X_A + X_T \\ m(\dot{v} + ru - pw) = Y_G + Y_A + Y_T \\ m(\dot{w} + pv - qu) = Z_G + Z_A + Z_T \\ I_{xx}\dot{p} - I_{xz}(\dot{r} + pq) - (I_{yy} - I_{zz})qr = \mathcal{L}_A + \mathcal{L}_T \\ I_{yy}\dot{q} - I_{xz}(r^2 - p^2) - (I_{zz} - I_{xz})rp = \mathcal{M}_A + \mathcal{M}_T \\ I_{zz}\dot{r} - I_{xz}(\dot{p} + qr) - (I_{xx} - I_{yy})pq = \mathcal{N}_A + \mathcal{N}_T \end{cases} \tag{2.1}$$

which express the 3 translational and 3 rotational degrees of freedom. The dependencies of the RHS terms on attitude and position parameters with respect to a fixed inertial frame make the use of auxiliary kinematic equations (Gimbal Equations and Navigation Equations) necessary:

$$\begin{cases} \dot{\phi} = p + (q \sin\phi + r \cos\phi)\frac{\sin\theta}{\cos\theta} \\ \dot{\theta} = q \cos\phi - r \sin\phi \\ \dot{\psi} = (q \sin\phi - r \cos\phi)/\cos\theta \end{cases} \tag{2.2}$$

$$\begin{Bmatrix} \dot{x}_{E,G} \\ \dot{y}_{E,G} \\ \dot{z}_{E,G} \end{Bmatrix} = [T(\phi, \theta, \psi)_{EB}] \begin{Bmatrix} \dot{x}_{E,G} \\ \dot{y}_{E,G} \\ \dot{z}_{E,G} \end{Bmatrix} \tag{2.3}$$

Introducing the state vector of motion:

$$|\mathrm{r} = [u, v, w, p, q, r, x_{E,G}, y_{E,G}, z_{E,G}, \phi, \theta, \psi]^T \tag{2.4}$$

and the input vector:

$$u = [\delta_{\boldsymbol{c}}, \delta_T, \delta_{\boldsymbol{a}}, \delta_{\boldsymbol{r}}] \tag{2.5}$$

the system can be written compactly:

$$\dot{\mathbf{x}} = f(\mathbf{u}, \mathbf{x}) \tag{2.6}$$

The system of 12 coupled differential equations written for the study of aircraft motion is strongly non-linear, and it is not simple to find an analytical integration method. Equations formulated in this way are not useful for understanding the link between geometric and aerodynamic properties and the stability and maneuverability properties of the aircraft. For this reason, linearized equations of motion are very interesting as they provide important qualitative and quantitative information about flight. It is possible to consider linearized equations valid only under the hypothesis of small perturbations around the initial balanced motion condition. In this exercise, a nominal condition of balanced, longitudinal-symmetric, translated motion at constant altitude is considered. The hypothesis of small perturbations allows us to linearize the system of equations of motion, arriving at a system of differential equations that can be studied with the classic methods of *linear time-invariant systems* (LTI) theory. The linear system obtained through Brayan's hypotheses can be studied by separating the longitudinal dynamics from the lateral-directional dynamics. In this context, a mention is made of the determination, onerous in terms of steps and treatment, of the system of linearized equations of longitudinal-symmetric motion. Rather, space and attention are dedicated to the numerical solution of the aforementioned system and to the determination and representation of the aircraft's response modes (short period and phugoid), and more generally to the study of the longitudinal free response.

2.2 Problem Setup

The linearization starts from a nominal condition x_N which is a solution of the equations with a certain initial condition x_0 and a time history input u_N, meaning it is a particular trajectory in the state space. We thus start from a trim condition such that the vector of the derivatives of the dynamic variables is identically null and where the motion is translated, symmetric, and with level wings:

- $p_0 = q_0 = r_0 = \beta = v_0 = \phi_0 = \psi_0 = 0$

Starting from the nominal condition, stability axes are introduced, and therefore it results:

- $\alpha_0 = 0$

if the hypothesis of constant altitude is also made, it results:

- $\theta_0 = \gamma_0 = 0$

Our interest focuses on the motion x(t) corresponding to small perturbations around the nominal condition x_N. It corresponds to a solution of the motion problem corresponding to different initial conditions, or a different control law compared to the nominal ones, and by the hypothesis of small perturbations, it will be expressed in the form:

$$x(t) = x_N(t) + \Delta x \tag{2.7}$$

The total values of state parameters can be substituted into the equations of motion as the sum of the values corresponding to the nominal conditions and the perturbations on the variable, for example:

$$U = U_0 + u \tag{2.8}$$

It is possible to treat the perturbation values in the equations thus obtained as infinitesimals and therefore neglect infinitesimal terms of order higher than the first. The linearized equations of motion are obtained where the unknowns are represented by the perturbations of the state parameters:

$$\begin{cases} m(\dot{u} + W_0 q + Q_0 w - R_0 v - V_0 r = \Delta X_G + \Delta X_A + \Delta X_T \\ m(\dot{v} + U_0 r + R_0 u - P_0 w - W_0 p = \Delta Y_G + \Delta Y_A + \Delta Y_T \\ m(\dot{w} + V_0 p + P_0 v - Q_0 u - U_0 q = \Delta Z_G + \Delta Z_A + \Delta Z_T \end{cases} \tag{2.9}$$

$$\begin{cases} I_{xx}\dot{p} - I_{xz}\dot{r} - I_{xz}(P_0 q + Q_0 p) - (I_{yy} - I_{zz})(Q_0 r - R_0 q) = \Delta\mathcal{L}_A + \Delta\mathcal{L}_T \\ I_{yy}\dot{q} - 2I_{xz}(P_0 q + Q_0 p) - (I_{zz} - I_{xx})(R_0 p - P_0 r) = \Delta\mathcal{M}_A + \Delta\mathcal{M}_T \\ I_{zz}\dot{r} - I_{xz}\dot{p} - I_{xz}(P_0 q + Q_0 p) - (I_{xx} - I_{yy})(P_0 q - Q_0 p) = \Delta\mathcal{N}_A + \Delta\mathcal{N}_T \end{cases} \tag{2.10}$$

Through Bryan's hypotheses, for which the variability of symmetric forces or moments with respect to asymmetric parameter perturbations (and vice versa) is negligible, it is evident that it is possible to separate longitudinal and lateral-directional dynamics. The system of equations for longitudinal-symmetric flight with fixed controls in matrix form assumes the following form:

$$\begin{bmatrix} \dot{x}_{\text{Lon}} \\ \dot{x}_{\text{LD}} \end{bmatrix} = \begin{bmatrix} A_{\text{Lon}} & 0 \\ 0 & A_{\text{LD}} \end{bmatrix} \begin{bmatrix} x_{\text{Lon}} \\ x_{\text{LD}} \end{bmatrix} + \begin{bmatrix} B_{\text{Lon}} & 0 \\ 0 & B_{\text{LD}} \end{bmatrix} \begin{bmatrix} u_{\text{Lon}} \\ u_{\text{LD}} \end{bmatrix} \tag{2.11}$$

A defines the system matrix, while B defines the input matrix. The two dynamics can therefore be analyzed with the tools of Linear Time-Invariant (LTI) dynamic systems theory. This implies that the stability of the aircraft dynamic system can be studied starting from the free response of the system with an assigned initial condition. Having identified the left and right eigenvectors of the system matrix A (indicated respectively ξ_k and χ_k) and their respective eigenvalues (λ_k), it can be shown that the generic free response given an initial condition $x0$ can be written as:

$$x(t) = \sum_{k=1}^{n_x} \xi_k^0 e^{\lambda_k t} \chi_k \tag{2.12}$$

More generally, the vector x_0 represents a perturbation of the initial longitudinal equilibrium conditions of the aircraft which excites a free response. The system mode is defined, apart from the real multiplicative factor $e^{\sigma_k t}$, as the vector $e_k^{\lambda_k t}$.

2.3 Longitudinal Dynamics

From the separation of the equations, it follows that it is possible to separate the eigenvalues of the motion into two distinct groups

$$\Delta(s) = (s - \lambda_1)(s - \lambda_6)\ldots(s - \lambda_7)\ldots(s - \lambda_{12}) \tag{2.13}$$

where the first 6 eigenvalues refer to longitudinal dynamics and the remaining to lateral-directional dynamics.

In this section, we study the longitudinal dynamics.

$$\Delta_{\text{Lon}}(s) = (s - \lambda_{\text{RANGE}})\,(s - \lambda_{\text{HEIGHT}})\,(s - \lambda_{\text{PH}})\left(s - \lambda_{\text{PH}}^*\right)(s - \lambda_{\text{SP}})\left(s - \lambda_{\text{SP}}^*\right) \tag{2.14}$$

The characteristic polynomial will be characterized by two real and distinct roots λ_{RANGE}, λ_{HEIGHT} associated with the linearized equations of the perturbation x_{EG} and z_{EG}. Typically, these two roots are negligible, and therefore their corresponding modes will not be considered. While the other two complex conjugate pairs of roots, λ_{PH} and λ_{SP}, represent two oscillatory modes: in particular, they characterize a *long-period* (or phugoid) mode and a *short-period* mode. In the case of the short-period mode, a much higher damping and natural frequency will be obtained than those obtained in the long period.

2.4 Longitudinal Characteristics of a Boeing 747, pure short-period and pure phugoid mode

In this section, a Matlab script is used to derive the A matrix by calculating eigenvalues and eigenvectors (implemented with the *eig* command), diagramming the modal responses, eigenvalues, and eigenvector phasors. The existence of short-period and phugoid modes is verified. The calculation code, as constructed, allows choosing one of the 5 conditions present in 2.1 through the choice of *condition*. In particular, the plots relate to condition 7. This configuration was chosen to highlight, as visible from 2.1, the *Tuck under*. This phenomenon was first noticed by pilots during World War II and shows a tendency for the aircraft to pitch down once M_{DD} is exceeded. In particular for the Boeing 747, it is evident that moving from condition 7 to condition 10, where there is an increase in *Mach* number at the same altitude, the sign of C_{MM} changes and becomes negative. The characteristics are reported below:

Listing 2.1

Condizione	2	5	7	9	10
h (ft)	SL	20000	20000	40000	40000
M	0,25	0,50	0,80	0,80	0,90
α_B (deg)	5,70	6,80	0,00	4,60	2,40
W (lbf)	564000	636640	636640	636640	636640
I_{yy} (slug ft^2)	$32{,}30 \cdot 10^6$	$33{,}10 \cdot 10^6$	$33{,}10 \cdot 10^6$	$33{,}10 \cdot 10^6$	$33{,}10 \cdot 10^6$
C_L	1,10	0,68	0,27	0,66	0,52
C_D	0,10	0,04	0,02	0,04	0,04
C_{L_α}	5,70	4,67	4,24	4,92	5,57
C_{D_α}	0,66	0,37	0,08	0,43	0,53
$C_{\mathcal{M}_\alpha}$	−1,26	−1,15	−0,63	−1,03	−1,61
$C_{L_{\dot\alpha}}$	6,70	6,53	5,99	5,91	5,53
$C_{\mathcal{M}_{\dot\alpha}}$	−3,20	−3,35	−5,40	−6,41	−8,82
C_{L_q}	5,40	5,13	5,01	6,00	6,94
$C_{\mathcal{M}_q}$	−20,80	−20,70	−20,50	−24,00	−25,10
C_{L_M}	0	−0,09	0,11	0,21	−0,28
C_{D_M}	0	0	0,01	0,03	0,24
$C_{\mathcal{M}_M}$	0	0,12	−0,12	0,17	−0,11
$C_{L_{\delta_e}}$	0,338	0,356	0,270	0,367	0,300
$C_{\mathcal{M}_{\delta_e}}$	−1,34	−1,43	−1,06	−1,45	−1,20

Figura 2.1 Analyzed flight conditions for a Boeing 747

```
clc; clear all; close all;
%%
    %% modifico qua in modo da poter considerare diverse condizioni
%%

%% Scelta della condizione di volo
condition = 3;
%% ac data
mass = [255753 2.8869e+05 2.8869e+05 2.8869e+05 2.8869e+05]; %vettore delle masse
mass= mass( condition ); %kg
Iyy = [4.38*10^7 4.2740*10^7 4.2740*10^7 4.2740*10^7 4.2740*10^7];% vettore momenti di inerzia
Iyy = Iyy ( condition ) ;
S = 510.97;%superificie alare
cbar = 8.32;% corda media aerodinamica
SM0 = 0.22;%margine statico di sicurezza iniziale
SM= 0.22 ;%(X_N - X_G) / cbar

%% Condizioni di volo
zEG_0 = [0 2e+4 2e+4 4e+4 4e+4]; %quote di volo
zEG_0 = zEG_0*0.3048;% Conversione f t  >m
zEG_0 = zEG_0( condition) ;
q0 = 0;% Velocit  angolare di beccheggio
gamma0 = 0;% Angolo di rampa
g = 9.81; %g
[~, a_0 ,~, rho0] = atmosisa ( zEG_0 ) ; %ISA
Mach0 = [0.25 0.5 0.8 0.8 0.9];% vettore Mach
Mach0 = Mach0( condition ) ;
U_0 = Mach0* a_0 ; %velocit
```

```
alfa_B_0 = [5.70 6.80 0.00 4.60 2.40] ;%vettore alpha body
alfa_B_0 = alfa_B_0( condition ) ;
qbar_0 = 0.5* rho0*U_0^2; %pressione dinamica
mu_0 = 2*mass/(rho0 *S*cbar ) ;

%Definizione dei vettori delle caratteristiche aerodinamiche e delle derivate
% di stabilit
C_L = [1.10 0.68 0.27 0.66 0.52];
C_L =C_L (condition);
C_D = [0.10 0.04 0.02 0.04 0.04];
C_D = C_D( condition ) ;
C_L_alpha = [5.70 4.67 4.24 4.92 5.57];
C_L_alpha =C_L_alpha (condition);
C_D_alpha = [0.66 0.37 0.08 0.43 0.53];
C_D_alpha=C_D_alpha(condition);
c_m_alpha_0 = [-1.26 -1.15 -0.63 -1.03 -1.61];
c_m_alpha_0=c_m_alpha_0(condition);
C_m_alpha = c_m_alpha_0 *SM/SM0;

C_L_alphadot = [6.70 6.53 5.99 5.91 5.53];
C_L_alphadot = C_L_alphadot( condition) ;
C_m_alphadot = [-3.20 3.35 -5.40 -6.41 -8.82];
C_m_alphadot= C_m_alphadot (condition ) ;

C_L_q = [5.40 5.13 5.01 6.00 6.94];
C_L_q = C_L_q( condition ) ;
C_m_q = [-20.80 -20.70 -20.50 -24 -25.10];
C_m_q = C_m_q( condition ) ;
C_L_Mach = [0 -0.09 0.11 0.21 -0.28] ;
C_L_Mach = C_L_Mach( condition ) ;
C_D_Mach = [0 0 0.01 0.03 0.24];
C_D_Mach = C_D_Mach( condition ) ;
C_m_Mach = [0 0.12 -0.12 0.17 -0.11] ;
C_m_Mach = C_m_Mach( condition ) ;
C_L_de = [0.338 0.356 0.270 0.367 0.300];
C_L_de = C_L_de ( condition ) ;
C_m_de = [-1.34 -1.43 1.06 -1.45 -1.20];
C_m_de = C_m_de ( condition ) ;

% Calcolo derivare di stabilit
% Deriv stab longitudinali
    X_u = -(qbar_0*S/(mass*U_0))*(2*C_D + Mach0*C_D_Mach); % Dipende da D e T quindi Cd ed M
    X_w = (qbar_0*S/(mass*U_0))*(C_L - C_D_alpha); % dipende da W->L->Cl
    Z_u = -(qbar_0*S/(mass*U_0))*(2*C_L + (Mach0^2/(1-Mach0^2))*C_L_Mach); % constant thrust
    Z_w = -(qbar_0*S/(mass*U_0))*(C_D + C_L_alpha); % Proporzionale a -CLalfa
    Z_wdot = -(1/(2*mu_0))*C_L_alphadot;
    Z_q = -(U_0/(2*mu_0))*C_L_q;
    M_u = (qbar_0*S*cbar/(Iyy*U_0))*Mach0*C_m_Mach;
    M_w = (qbar_0*S*cbar/(Iyy*U_0))*C_m_alpha; % dipende da W->L->Cm_alpha
```

```
    M_wdot = (rho0*S*(cbar^2)/(4*Iyy))*C_m_alphadot; % Proviene dal downwash
    M_q = (rho0*U_0*S*(cbar^2)/(4*Iyy))*C_m_q; % Proporzionale a CMq (derivata di smorzamento)(-)
    k_hat = M_wdot/(1-Z_wdot);




% Costruzione matrice  A_LON
    A = NaN(4);
    A(1,1) = X_u;
    A(1,2) = X_w;
    A(1,3) = 0;
    A(1,4) = -g*cos(gamma0);
    A(2,1) = Z_u/(1-Z_wdot);
    A(2,2) = Z_w/(1-Z_wdot);
    A(2,3) = (Z_q + U_0)/(1-Z_wdot);
    A(2,4) = -g*sin(gamma0)/(1-Z_wdot);
    A(3,1) = M_u + k_hat*Z_u;
    A(3,2) = M_w + k_hat*Z_w;
    A(3,3) = M_q + k_hat*(Z_q + U_0);
    A(3,4) = -k_hat*g*sin(gamma0);
    A(4,1) = 0;
    A(4,2) = 0;
    A(4,3) = 1;
    A(4,4) = 0;

    % Derivate di controllo
    X_dT = 0;
    Z_dT = 0;
    M_dT = 0;
    X_de = 0;
    Z_de = -qbar_0*S/mass*C_L_de;
    M_de = qbar_0*S*cbar/Iyy*C_m_de;

% RIPOSTA LIBERA E CARATTERISTICHE MODALI
% determinazione autovalori e autovettori
% Troviamo autovalori ed autovettori del sistema: V ci restituisce una
% matrice di autovalori e D la matrice dei corrispondenti autovettori tale
% che A_lon * V = V * D
    [V,D] = eig(A);

% Per avere una scala dei valori si dividono tutte le componenti per la
% quarta componente tale che V_SP(4) = V_fugoide(4) = 1 + 0*i.
% I valori della seconda e quarta colonna non vengono considerati solo
% perch  sono semplicemente i complessi coniugati ripettivamente della
% prima e terza colonna

% Autovalori corto periodo rispetto a theta (4)
    V_SP = V(:,1);
    V_SP = V_SP/V_SP(4);
    V_SP(1) = V_SP(1)/U_0;
    V_SP(2) = V_SP(2)/U_0;
    V_SP(3) = V_SP(3)/(2*U_0/cbar);
```

```

% Autovalori lungo periodo rispetto a theta (4)
    V_Ph = V(:,3);
    V_Ph = V_Ph/V_Ph(4);
    V_Ph(1) = V_Ph(1)/U_0;
    V_Ph(2) = V_Ph(2)/U_0;
    V_Ph(3) = V_Ph(3)/(2*U_0/cbar);


% SS ANALIZZA SPAZIO DEGLI STATI (PER SISTEMA LINEARE TEMPO INVARIANTE)
% Definizione matrice
    sys = ss( ...
    A, ...              % A
    zeros(4,1), ...     % B
    eye(4,4), ...       % C
    zeros(4,1) ...      % D
    );

% CORTO PERIODO E FUGOIDE
    V1SP = V(:,1);
    V1SP   = V1SP/V1SP(4);
    V1PH = V(:,3);
    V1PH   = V1PH/V1PH(4);

% Valori iniziali
    X0_1 = real(V1SP);
    X0_2 = real(V1PH);

% Risuluzione problema LTI ai valori iniziali
% (INITIAL) VALUTA SOLUZIONI PROBLEMA (AUTOVALORI)
    [Y_1,T_1,X_1] = initial(sys,X0_1);
    [Y_2,T_2,X_2] = initial(sys,X0_2);


% ESERCIZIO 16.2
```

The values are inserted into the matrices of the LTI system. Subsequently, using the *symbolic tool*, the eigenvalues and eigenvectors are calculated. Finally, the characteristic quantities of the *short period* and *long period* motions were calculated.

It is important to note that the phasor representation was done by normalizing with respect to the fourth component, i.e., with respect to θ, which will have a modulus component equal to 1. In the code, the matrix $\mathbf{T_{LON}}$ has been introduced, which contains on the diagonal the correctly non-dimensionalized and normalized perturbations.

$$\mathbf{T_{LON}} = \begin{bmatrix} \dfrac{1}{U_0} & 0 & 0 & 0 \\ 0 & \dfrac{1}{U_0} & 0 & 0 \\ 0 & 0 & \dfrac{\bar{c}}{2U_0} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

To obtain the responses of only short period and only phugoid, assign initial condi-

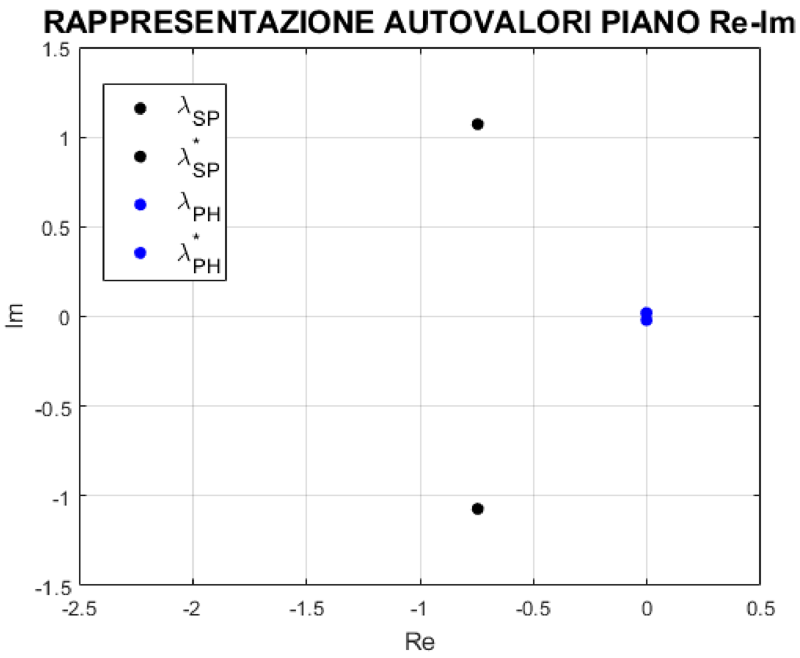

Figura 2.2 Representation of eigenvalues in the complex plane. Two pairs of complex conjugate eigenvalues

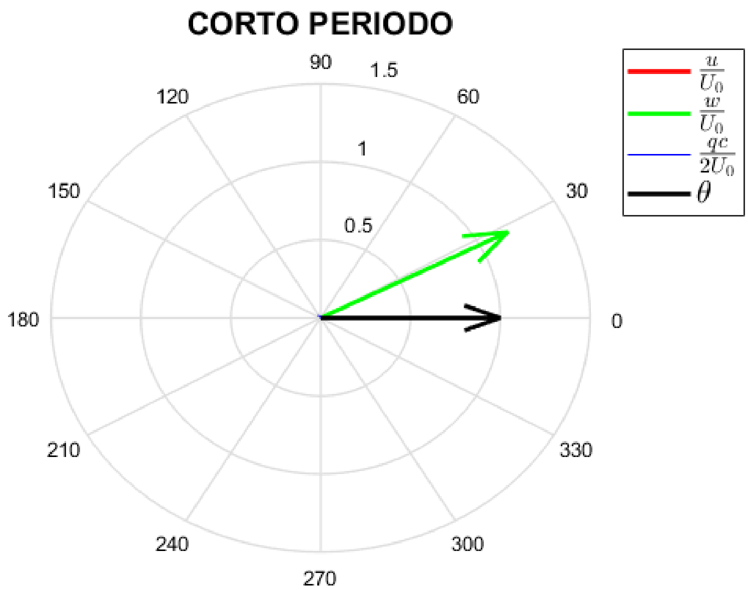

Figura 2.3 Phasor diagram in the complex plane. Short Period

tions coinciding with the eigenvector λ_1 (short period) and λ_3 (phugoid). The following characteristics are also calculated via the Matlab script:

- Damping coefficient ζ;
- Natural frequency ω_n;
- Period T;
- Half-life time $t_{\frac{1}{2}}$;
- Number of cycles for half-life $t_{\frac{1}{2}}$.

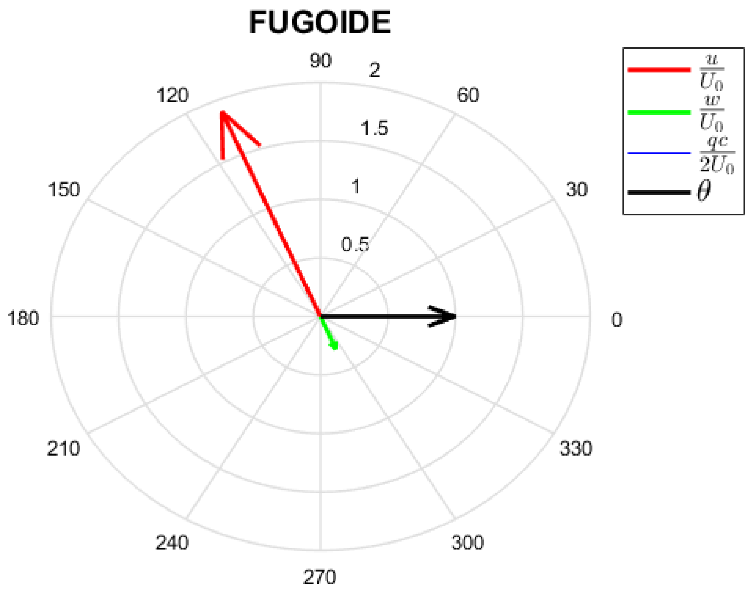

Figura 2.4 Phasor diagram in the complex plane. Phugoid

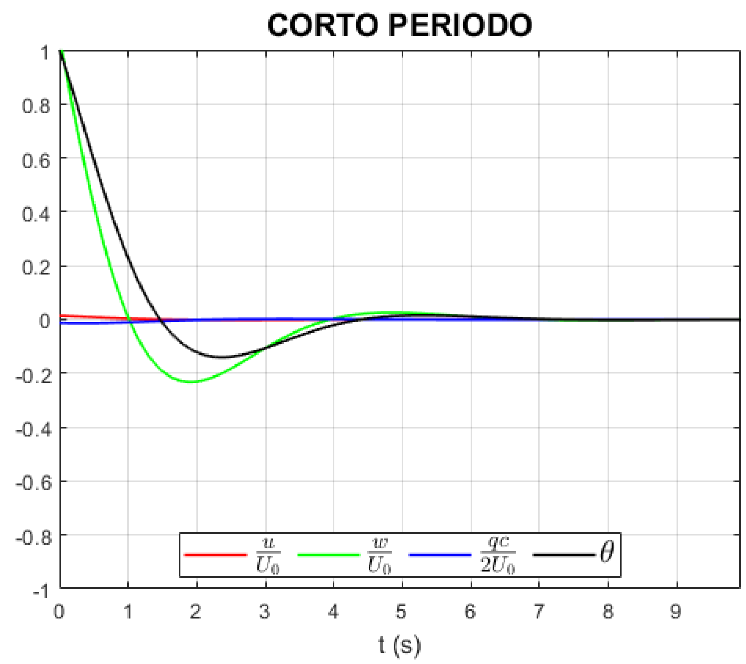

Figura 2.5 Free response of the Boeing 747 aircraft to a longitudinal perturbation for flight condition 7 defined in 2.1. The response was obtained by exciting only the short-period mode.

Approximate formulas for short and long period are also used. The results are presented:

It should be noted that the approximate models of the phugoid motion do not provide particularly accurate values for the damping coefficient, while they are more accurate in approximating the natural frequency.

The natural frequency and damping coefficient are well approximated by the short period approximation formulas. With the short period coarse approximation formulas, on the other hand, the natural frequency is calculated with reasonable approximation, while the damping coefficient is underestimated.

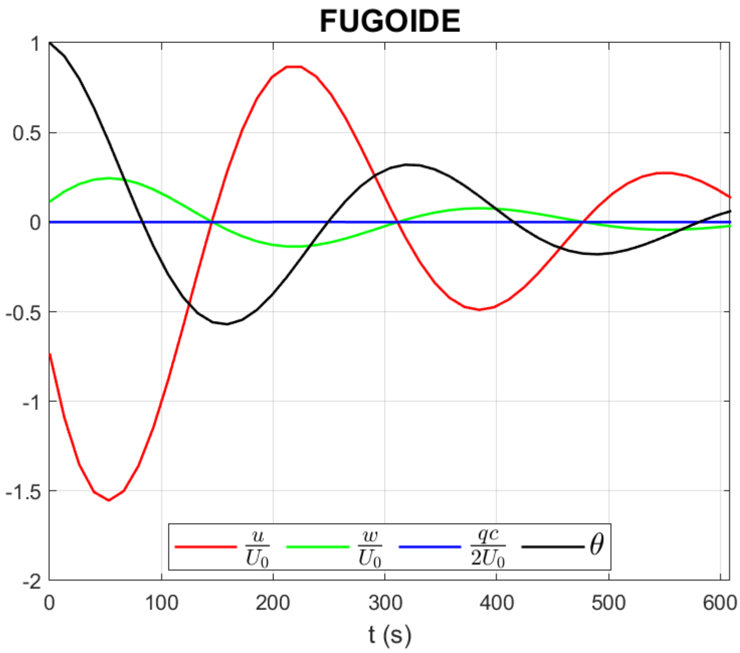

Figura 2.6 Free response of the Boeing 747 aircraft to a longitudinal perturbation for flight condition 7 defined in 2.1. The response was obtained by exciting only the phugoid mode.

```
FUGOIDE
smorzamento 0.18062
frequenza 0.019237 s^-1
durata 332.0745 s
tempo di dimezzamento 199.4854 s
numero di cicli di dimezzamento 0.60072 s
```

Figura 2.7 Phugoid: exact values

2.5 Longitudinal Dynamics with Forcing Perturbation/ Comparison with Q7

It is interesting to examine the behavior of the system when the response is not free, but is instead subject to forcings, indicated by the vector $\mathbf{u}_{\text{lon}}$ which is not null. Therefore, in the next paragraph, we will address the following problem.

$$\begin{cases} \dot{\mathbf{x}}_{\text{LON}}(0) = \mathbf{A}_{\text{LON}}\mathbf{x}_{\text{LON}} + \mathbf{B}_{\text{LON}}\mathbf{u}_{\text{LON}} \\ \mathbf{x}_{\text{LON}}(0) = 0 \\ \mathbf{u}_{\text{LON}} = [\delta_e(t), 0]^T \qquad\qquad for 0 \leq t < t_f \end{cases} \tag{2.16}$$

The matrices that make up the system must be constructed, and then a law of motion for the elevator $\delta_e(t)$ must be assigned. In particular, a *doublet* type law has been assigned, and the *lsim* function is used. This exercise also has the further purpose of comparing 2 different models, that of notebook 7 and that of notebook 16. The calculation code is reported below:

Listing 2.2

```
CARATTERISTICHE APPROSSIMATE DI FUGOIDE
smorzamento 0.24066
frequenza 0.019165 s^-1
durata 337.7689 s
tempo di dimezzamento  2.1818s
numero di cicli di dimezzamento 0.0064596 s
CARATTERISTICHE APPROSSIMATE DI FUGOIDE "coarse"
smorzamento 0.054289
frequenza 0.06456 s^-1
durata 97.4668 s
tempo di dimezzamento  9.6719s
numero di cicli di dimezzamento 0.099233 s
```

Figura 2.8 Phugoid: approximate values

```
CORTO PERIODO
smorzamento 0.57162
frequenza 1.3058 s^-1
durata 5.8644 s
tempo di dimezzamento 0.92865s
numero di cicli di dimezzamento 0.15835 s
```

Figura 2.9 Short period: exact values

```

global myAC g rho0 q0 gamma0 z0 deltas0 deltae0 deltaT0 delta_e delta_T delta_s

%% Input properties
aircraftDataFileName = 'DSV_Aircraft_data.txt';
myAC = DSVAircraft(aircraftDataFileName);
g = 9.81;

%% Initial Trim Conditions
aircraftDataFileName = 'DSV_Aircraft_data.txt';
myAC = DSVAircraft(aircraftDataFileName);
tf = 300;

if (myAC.err == -1)
    disp('Termination.')
else
    % Constants and initial conditions
    xEG_0 = 0; % [m]
    zEG_0 = -4000; % Altitude [m]
    q0 = convangvel(0.000, 'deg/s', 'rad/s'); % Pitch angular velocity
    gamma0 = convang(0.00, 'deg', 'rad'); % Ramp angle
    [air_Temp0, sound_speed0, air_pressure0, rho0] = atmosisa(-zEG_0);
    v0 = 0.5 * sound_speed0; %Mach 0.5

    % Process of cost function minimization
    x0 = [0; 0; 0; 0.5]; % Initial guess for the design vector
    Aeq = zeros(4);
```

```
CARATTERISTICHE APPROSSIMATE DI CORTO PERIODO
frequenza 1.3201 s^-1
smorzamento 0.57056
durata 5.7956 s
tempo di dimezzamento 0.92028 s
numero di cicli di dimezzamento 0.15879 s

CARATTERISTICHE APPROSSIMATE DI CORTO PERIODO "coarse"
frequenza 1.1433 s^-1
smorzamento 0.30608
durata 5.7725 s
tempo di dimezzamento 1.9807 s
numero di cicli di dimezzamento 0.34176 s
```

Figura 2.10 Short period: approximate values

```matlab
Aeq(3, 3) = 1;
delta_s_0 = convang(-1.000, 'deg', 'rad');
beq = zeros(4, 1);
beq(3, 1) = delta_s_0;%ho fissato delta_s_0

lb = [convang(-15, 'deg', 'rad'), convang(-20, 'deg', 'rad'), convang(-5, 'deg', 'rad'), 0.2];
ub = [convang(15, 'deg', 'rad'), convang(13, 'deg', 'rad'), convang(5, 'deg', 'rad'), 1.0];

options = optimset('tolfun', 1e-9, 'Algorithm', 'interior-point');

global V_0 q_0 gamma_0 rho_0
V_0 = v0;
q_0 = q0;
gamma_0 = gamma0;
rho_0 = rho0;

[x, fval] = fmincon(@(x) costLongEquilibriumStaticStickFixed(x), x0, ...
    [], [], Aeq, beq, lb, ub, @myNonLinearConstraint, options);

alpha0 = x(1);
alpha_0_deg = convang(x(1), 'rad', 'deg');
delta_e_0 = x(2);
deltae0_deg = convang(x(2), 'rad', 'deg');
deltas0 = x(3);
delta_s_0_deg = convang(x(3), 'rad', 'deg');
deltaT0 = x(4);
theta0 = alpha0 + gamma0 - myAC.mu_x;
z0 = -4000;
xE0 = 0;
[~, ~, ~, rho0] = atmosisa(-z0);

vBreaksdelta_e(1, :) = [0, 4.99, 5, 6.0, 6.01, 14.99, 15, 16.0, 16.01, tf];
vBreaksdelta_e(2, :) = [deltae0_deg, deltae0_deg, deltae0_deg-1,...
```

```
        deltae0_deg-1, deltae0_deg, deltae0_deg, deltae0_deg+1,
    deltae0_deg+1, ...
        deltae0_deg, deltae0_deg];

    delta_e_deg = @(t) interp1(vBreaksdelta_e(1, :), vBreaksdelta_e(2, :)
    , t, 'linear');
    delta_e = @(t) convang(delta_e_deg(t), 'deg', 'rad');
    delta_T = @(t) interp1 ([0 tf], [deltaT0 deltaT0], t);
    delta_s = @(t) interp1 ([0 tf], [deltas0 deltas0], t);
    t_span=linspace(0,tf,1000); %vettore dei tempi
    x0 = [v0, alpha0, q0, xE0, z0, theta0];%condizioni iniziali
    options = odeset('RelTol', 1e-3, 'AbsTol', 1e-3 * ones(1, 6));
    [vTime, mState] = ode45(@eqLongDynamicStickFixed, t_span, x0);
    vdeltae = delta_e_deg(vTime);
end

%%%%%%%%%Parte linearizzata%%%%%%%%%
h = -zEG_0;
Mach_0 = .5; %Mach
alpha_b = alpha_0_deg;
% h_0 = convlength(h,'ft','m');
U_0 = Mach_0 * sound_speed0*cos(convang(alpha_b,'deg','rad')); %   uguale
    alla V_0 di prima
%
S = myAC.S; % m^2
cbar = myAC.mac; % m
Weight_0 = myAC.W; % N
mass = Weight_0/g;
Iyy_0 = mass*myAC.k_y^2;
Iyy = Iyy_0;

C_L = 2*Weight_0/(rho0*U_0^2*S);
C_D = myAC.CD_0 + myAC.K*(C_L)^myAC.m;

C_L_alpha = myAC.CL_alpha; % 1/rad
C_D_alpha = 2*myAC.CL_alpha*alpha0; %Tramite la polare
C_m_alpha_0 = myAC.Cm_alpha; % 1/rad
C_L_de = myAC.CL_delta_e; % 1/rad
C_m_de = myAC.Cm_delta_e; % 1/rad
C_L_alphadot = myAC.CL_alpha_dot; % 1/rad
C_m_alphadot = myAC.Cm_alpha_dot; % 1/rad
C_L_q = myAC.CL_q; % 1/rad
C_m_q = myAC.Cm_q; % 1/rad
C_L_Mach = 0 ; %trascurato
C_D_Mach = 0; %trascurato
C_m_Mach = 0.10; %valore plausibile

%% Data elaboration
% KV = 0; CTFIX = 0;
%% Constants Computation -----> Quaderno 16, pg. 60
qbar_0 = 0.5*rho0*(U_0^2);
mu_0 = mass/(0.5*rho0*S*cbar);
Gamma_0 = 0.0; % rad
SM_0 = 0.22;
SM = 0.22;
C_m_alpha = C_m_alpha_0*(SM/SM_0); % 1/rad
```

```
X_u = -(qbar_0*S/(mass*U_0))*(2*C_D + Mach_0*C_D_Mach); % constant thrust
X_w = (qbar_0*S/(mass*U_0))*(C_L - C_D_alpha);
X_wdot = 0; X_q = 0;
X_de = 0;
X_dT = qbar_0*S/mass*(0 + 0/U_0^2); %
Z_u = -(qbar_0*S/(mass*U_0))*( ...
    2*C_L + (Mach_0^2/(1-Mach_0^2))*C_L_Mach); % constant thrust
Z_w = -(qbar_0*S/(mass*U_0))*(C_D + C_L_alpha); Z_wdot = -(1/(2*mu_0))*C_L_alphadot;
Z_q = -(U_0/(2*mu_0))*C_L_q; Z_de = -qbar_0*S/mass*C_L_de;
Z_dT = 0;
%
M_u = (qbar_0*S*cbar/(Iyy*U_0))*Mach_0*C_m_Mach; M_w = (qbar_0*S*cbar/(Iyy*U_0))*C_m_alpha;
M_wdot = (rho0*S*(cbar^2)/(4*Iyy))*C_m_alphadot; M_q = (rho0*U_0*S*(cbar^2)/(4*Iyy))*C_m_q;
M_de = qbar_0*S/Iyy*C_m_de; M_dT = 0;
%
% Z_de =Z_de*100; M_de = M_de*100;
k_hat = M_wdot/(1-Z_wdot);
% Plant matrix --> cfr. (16.147b)
A_lon(1,1) = X_u;
A_lon(1,2) = X_w; A_lon(1,3) = 0;
A_lon(1,4) = -g*cos(Gamma_0); A_lon(2,1) = Z_u/(1 - Z_wdot);
A_lon(2,2) = Z_w/(1 - Z_wdot); A_lon(2,3) = (Z_q + U_0)/(1 - Z_wdot);
A_lon(2,4) = -g*sin(Gamma_0)/(1 - Z_wdot); A_lon(3,1) = M_u + k_hat*Z_u;
A_lon(3,2) = M_w + k_hat*Z_w; A_lon(3,3) = M_q + k_hat*(Z_q+U_0);
A_lon(3,4) = -k_hat*g*sin(Gamma_0); A_lon(4,1) = 0;
A_lon(4,2) = 0; A_lon(4,3) = 1;
A_lon(4,4) = 0;
%
%chiamo eig che prende in ingresso la matrice quadrata e resittuisce una
%matrice di autovalori V e autovalori D posi sulla diagonale
[V,D] = eig(A_lon);
%
W = inv(V); %ha gli autovalori sulle righe
% X_de/mass, X_dT/mass; ...
% Z_de/(mass-Z_wdot), Z_dT/(mass-Z_wdot); ...
% (M_de + Z_de*M_wdot/(mass-Z_wdot))/Iyy, ...
% (M_dT + Z_dT*M_wdot/(mass-Z_wdot))/Iyy; ...
% 0, 0];
B_lon = [ X_de , X_dT; ...
    Z_de/(1-Z_w) , Z_dT/(1-Z_w); ...
    M_de + k_hat* Z_de , M_dT + k_hat* Z_dT ; ...
    0, 0];
% Output matrices
C_lon = eye(3,4);
D_lon = zeros(3,2);
% make a state-space representation
sys_lon = ss( ...
    A_lon , ... % A
    B_lon , ... % B
    eye(4,4), ... % C
    zeros(4,2) ... % D
    );
% % Time vector
```

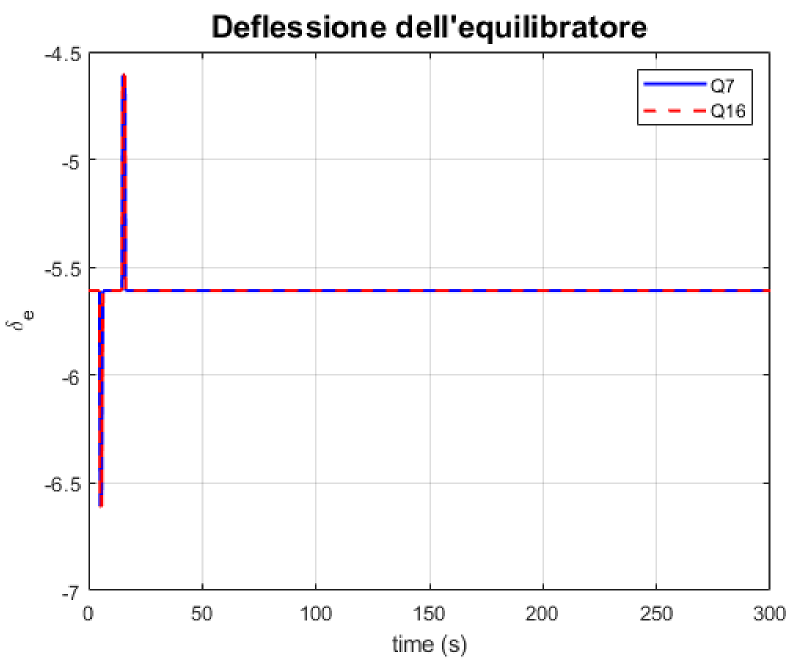

Figura 2.11 Elevator deflection laws

```
t_BP = [0, 4.99, 5, 6.0, 6.01, 14.99, 15, 16.0, 16.01,tf];
de_deg_BP = [0, 0 , -1, -1 , 0 , 0 , 1, 1, 0 , 0];

% figure; plot(t_BP, de_deg_BP); % Time vector
t_Elevator_Doublet = linspace(0,tf,1000)';

% Commanded deflection column vector

de_rad_Elevator_Doublet = interp1(t_BP,convang(de_deg_BP,'deg','rad'),
    t_Elevator_Doublet);
de_deg_Elevator_Doublet = interp1(t_BP,de_deg_BP,t_Elevator_Doublet); %
    Input u, nx2 matrix

u_Elevator_Doublet = [ ...
    de_rad_Elevator_Doublet, zeros(length(de_rad_Elevator_Doublet),1)];
```

The linear model of the equations is an approximate model that works better the smaller the disturbances. We see that with the linear model, smaller amplitudes and higher frequencies and damping are predicted for velocity and pitch angle.

2.6 Eigenvalues of Longitudinal Dynamics with Varying SM

Imagine varying the surface area S_h of the horizontal tailplane. Keeping l_h and all other parameters constant, varying this changes the stability margin SM. An array of SM values is created, corresponding to which the static pitch stability derivative $C_{M\alpha}$ is calculated. The following assumptions are made:

- the lift variability generated with S_h is neglected
- η_h is constant

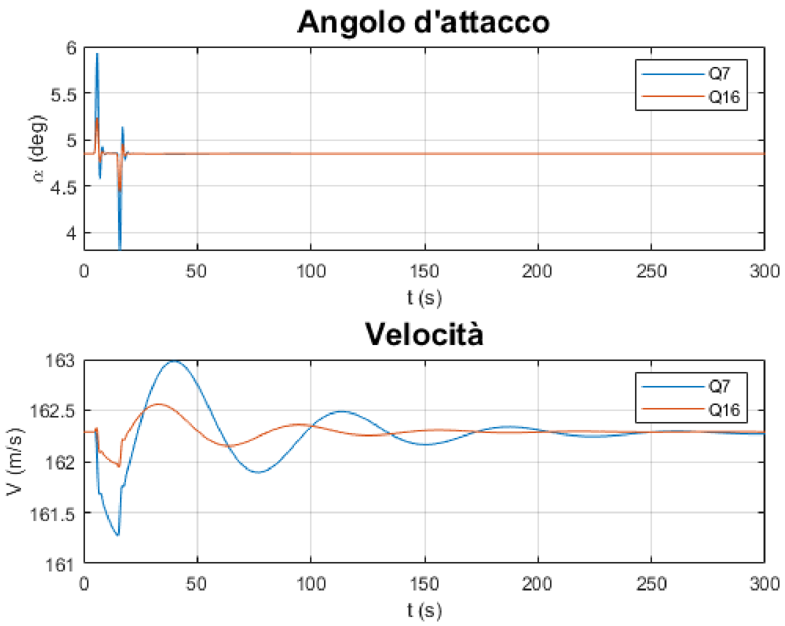

Figura 2.12 Variations of V and α

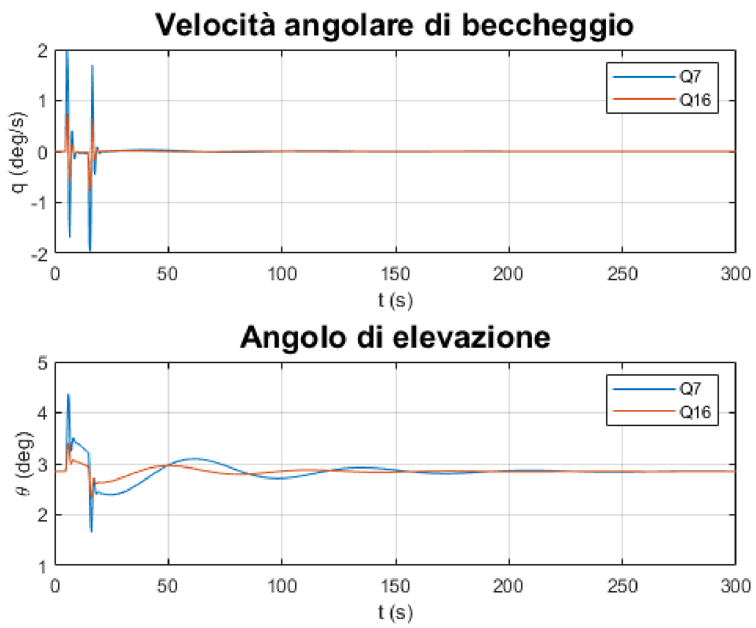

Figura 2.13 Variations of q and θ

- the downwash at the tail does not vary with S_h

The code is reported below.

Listing 2.3

```
clc; clear; clear all;
% Grandezze geometriche
condition = 3;
if condition==3
SM_vec=[0.22 , 0.15 0.10 , 0.05 , 0.035 , 0.025 , 0.0158 , 0.008 , ...
    0.005 , 0.0021 , 0.0 , -0.002 , -0.005 , -0.008 , -0.0145];
lvec=length(SM_vec);
```

```
elseif condition==4
    SM_vec=[0.22 , 0.15 0.10 , 0.05 , 0.035 , 0.025 , 0.0258]% , 0.008 ,
        ...
    %0.005 , 0.0021 , 0.0 , -0.002 , -0.005 , -0.008 , -0.0145];
    lvec=length(SM_vec);
    elseif condition==1
    SM_vec=[0.22 , 0.15 0.10 , 0.05 , 0.035 , 0.025 , 0.0258]% , 0.008 ,
        ...
    %0.005 , 0.0021 , 0.0 , -0.002 , -0.005 , -0.008 , -0.0145];
    lvec=length(SM_vec);
    elseif condition==2
    SM_vec=[0.22 , 0.15 0.10 , 0.05 , 0.035 , 0.025 , 0.0258]% , 0.008 ,
        ...
    %0.005 , 0.0021 , 0.0 , -0.002 , -0.005 , -0.008 , -0.0145];
    lvec=length(SM_vec);
elseif condition==5
    SM_vec=[0.22 , 0.15 0.10 , 0.05 , 0.035 , 0.025 , 0.0258]% , 0.008 ,
        ...
    %0.005 , 0.0021 , 0.0 , -0.002 , -0.005 , -0.008 , -0.0145];
    lvec=length(SM_vec);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% ac data
mass = [255753 2.8869e+05 2.8869e+05 2.8869e+05 2.8869e+05]; %vettore
    delle masse
mass= mass( condition ); %kg
Iyy = [4.38*10^7 4.2740*10^7 4.2740*10^7 4.2740*10^7 4.2740*10^7];%
    vettore momenti di inerzia
Iyy = Iyy ( condition ) ;
S = 510.97;%superficie alare
cbar = 8.32;% corda media aerodinamica
SM_0 = 0.22;%margine statico di sicurezza iniziale
%% Condizioni di volo
zEG_0 = [0 2e+4 2e+4 4e+4 4e+4]; %quote di volo
zEG_0 = zEG_0*0.3048;% Conversione f t  >m
zEG_0 = zEG_0( condition) ;
q0 = 0;% Velocit  angolare di beccheggio
Gamma_0 = 0;% Angolo di rampa
g_0 = 9.81; %g
[~, a_0 ,~, rho_0] = atmosisa ( zEG_0 ) ; %ISA
Mach_0 = [0.25 0.5 0.8 0.8 0.9];% vettore Mach
Mach_0 = Mach_0( condition ) ;
U_0 = Mach_0* a_0 ; %velocit
alfa_B_0 = [5.70 6.80 0.00 4.60 2.40] ;%vettore alpha body
alfa_B_0 = alfa_B_0( condition ) ;
qbar_0 = 0.5* rho_0*U_0^2; %pressione dinamica
mu_0 = 2*mass/(rho_0 *S*cbar ) ;

%Definizione dei vettori delle caratteristiche aerodinamiche e delle
    derivate
% di stabilit
C_L = [1.10 0.68 0.27 0.66 0.52];
C_L =C_L (condition);
C_D = [0.10 0.04 0.02 0.04 0.04];
```

```
C_D = C_D( condition ) ;
C_L_alpha = [5.70 4.67 4.24 4.92 5.57];
C_L_alpha =C_L_alpha (condition);
C_D_alpha = [0.66 0.37 0.08 0.43 0.53];
C_D_alpha=C_D_alpha(condition);
C_m_alpha_0 = [-1.26 -1.15 -0.63 -1.03 -1.61];
C_m_alpha_0=C_m_alpha_0(condition);


C_L_alphadot = [6.70 6.53 5.99 5.91 5.53];
C_L_alphadot = C_L_alphadot( condition) ;
C_m_alphadot = [-3.20 3.35 -5.40 -6.41 -8.82];
C_m_alphadot= C_m_alphadot (condition ) ;

C_L_q = [5.40 5.13 5.01 6.00 6.94];
C_L_q = C_L_q( condition ) ;
C_m_q = [-20.80 -20.70 -20.50 -24 -25.10];
C_m_q = C_m_q( condition ) ;
C_L_Mach = [0 -0.09 0.11 0.21 -0.28] ;
C_L_Mach = C_L_Mach( condition ) ;
C_D_Mach = [0 0 0.01 0.03 0.24];
C_D_Mach = C_D_Mach( condition ) ;
C_m_Mach = [0 0.12 -0.12 0.17 -0.11] ;
C_m_Mach = C_m_Mach( condition ) ;
C_L_de = [0.338 0.356 0.270 0.367 0.300];
C_L_de = C_L_de ( condition ) ;
C_m_de = [-1.34 -1.43 1.06 -1.45 -1.20];
C_m_de = C_m_de ( condition ) ;



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Derivate di stabilit

     X_u = -(qbar_0*S/(mass*U_0))*(2*C_D + Mach_0*C_D_Mach);
       % Dipende da D e T quindi Cd ed M
    X_w = (qbar_0*S/(mass*U_0))*(C_L - C_D_alpha);
     % dipende da W->L->Cl
    Z_u = -(qbar_0*S/(mass*U_0))*(2*C_L + (Mach_0^2/(1-Mach_0^2))*
   C_L_Mach); % constant thrust
    Z_w = -(qbar_0*S/(mass*U_0))*(C_D + C_L_alpha);
     % Proporzionale a -CLalfa
    Z_wdot = -(1/(2*mu_0))*C_L_alphadot;
    Z_q = -(U_0/(2*mu_0))*C_L_q;
    M_u = (qbar_0*S*cbar/(Iyy*U_0))*Mach_0*C_m_Mach;

    M_wdot = (rho_0*S*(cbar^2)/(4*Iyy))*C_m_alphadot;
      % Proviene dal downwash
    M_q = (rho_0*U_0*S*(cbar^2)/(4*Iyy))*C_m_q;
      % Proporzionale a CMq (derivata di smorzamento)(-)
    k_hat = M_wdot/(1-Z_wdot);



    %%%%%%%%% messo qua il ciclo
    for i=1:lvec
```

```
    SM=SM_vec(i);
    C_m_alpha=C_m_alpha_0*(SM/SM_0);

    M_w    = (qbar_0*S*cbar/(Iyy*U_0))... % [1/(ms)]
        *C_m_alpha;


    A_lon(1,1) = X_u;
    A_lon(1,2) = X_w;
    A_lon(1,3) = 0;
    A_lon(1,4) = -g_0*cos(Gamma_0);             % Prima riga

    A_lon(2,1) = Z_u/(1 - Z_wdot);
    A_lon(2,2) = Z_w/(1 - Z_wdot);
    A_lon(2,3) = (Z_q + U_0)/(1 - Z_wdot);
    A_lon(2,4) = -g_0*sin(Gamma_0)/(1 - Z_wdot); % Seconda riga

    A_lon(3,1) = M_u + k_hat*Z_u;
    A_lon(3,2) = M_w + k_hat*Z_w;
    A_lon(3,3) = M_q + k_hat*(Z_q+U_0);
    A_lon(3,4) = -k_hat*g_0*sin(Gamma_0);       % Terza riga

    A_lon(4,1) = 0;
    A_lon(4,2) = 0;
    A_lon(4,3) = 1;
    A_lon(4,4) = 0;                             % Quarta riga
    [V,D] = eig(A_lon);
    W = inv(V);
    eigen_vals_vec{i,1}= D ;
    eigen_vals_vec{i,2}= D ;
end
for i=1:lvec
    lambda_SP_arr(i)=eigen_vals_vec{i,1}(1 ,1);
    lambda_SP2_arr(i)=eigen_vals_vec{i,1}(2,2);
    sigma_SP_arr(i)=real(lambda_SP_arr(i));
    omega_SP_arr(i)=imag(lambda_SP_arr(i));
    sigma_SP2_arr(i)=real(lambda_SP2_arr(i));
    omega_SP2_arr(i)=imag(lambda_SP2_arr(i));

    lambda_PH_arr(i)=eigen_vals_vec{i,1}(3,3);
    lambda_PH2_arr(i)=eigen_vals_vec{i,1}(4,4);
    sigma_PH_arr(i)=real(lambda_PH_arr(i));
    omega_PH_arr(i)=imag(lambda_PH_arr(i));
    sigma_PH2_arr(i)=real(lambda_PH2_arr(i));
    omega_PH2_arr(i)=imag(lambda_PH2_arr(i));
end
```

2.7 Analysis of Aircraft Lateral-Directional Dynamics

The objective of the following paragraph is the analysis of the aircraft's lateral-directional dynamics, i.e., the evolution of small perturbations of non-symmetric variables around a nominal condition of balanced motion. In this discussion, we omit the determination

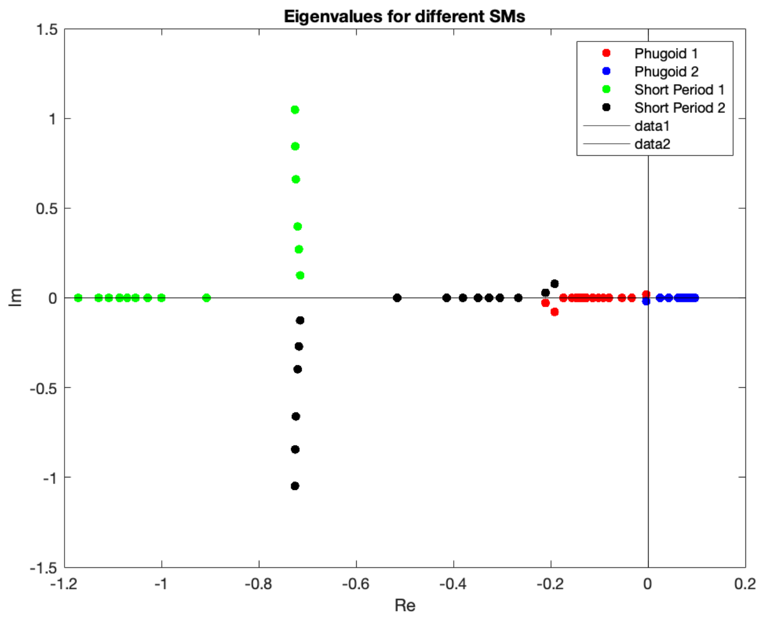

Figura 2.14 Variation of eigenvalue positions on the complex plane with varying SM

of the system of linearized equations for lateral-directional motion, focusing instead on deepening the numerical solution of such a system.

We focus our attention on determining and representing the aircraft's response modes, including the spiral mode, roll mode, and dutch roll. In particular, we will concentrate on the numerical solution of the system of differential equations describing the dynamics, exploring the details of each response mode.

The study extends to evaluating the aircraft's free response, providing a complete picture of its dynamic characteristics. The adopted approach allows for a thorough understanding of the aircraft's stability and maneuverability. The system of linearized equations for lateral-directional motion is presented in the following form:

$$x_{LD} = A_{LD}\, x_{LD} + B_{LD}\, u_{LD} \tag{2.17}$$

The roots of the characteristic polynomial of A_{LD} represent the eigenvalues of the system and can be expressed as:

$$\Delta_{\mathrm{LD}}(s) = (s - \lambda_{\mathrm{CrossRange}})\,(s - \lambda_{\mathrm{Heading}})(s - \lambda_{\mathrm{Spiral}})\;(s - \lambda_{\mathrm{Roll}})\;(s^2 + 2\zeta_{\mathrm{DR}}\,\omega_{\mathrm{n,DR}}\,s + \omega^2_{\mathrm{n,DR}}) \tag{2.18}$$

The four roots of the aircraft's lateral-directional dynamic system are associated with the linearized kinematic equations of the perturbations y_{EG} (transverse coordinate) and V' (heading angle). This association gives them the names of roots $\lambda_{Crossrange}$ and $\lambda_{Heading}$, respectively. They are characterized by real values and have values close to 0, for this reason the modes associated with these roots are considered irrelevant. The two roots, known as λ_{Spiral} and λ_{Roll}, are typically negative and also real. These roots are associated with the modal components recognized as the spiral mode and the roll mode. Finally, the complex root λ_{DR} represents an oscillatory mode called *dutch roll*. The objective now is to calculate eigenvalues and eigenvectors and then diagram the modal responses. The code is reported below.

Listing 2.4

```
t_fin = 100;
% Inerzie
Weight = convforce(636640,'lbf','N');
g_0    = 9.81;
mass   = Weight/g_0;
S       = 5500*(convlength(1,'ft','m'))^2;% Superficie alare
h=0;
[T_0, a_0, P_0, rho_0]=atmosisa(h);
cbar    = convlength(27.3,'ft','m');       % mac                 [m]
mu_0    = mass/(0.5*rho_0*S*cbar);
Gamma_0 = 0.0;                             % angolo di salita iniziale [rad]
AR=9.45;
b=sqrt(S*AR);
M=0.25;
U_0=a_0*M;
qbar_0  = 0.5*rho_0*(U_0^2);               % pressione dinamica iniziale [N/m^2]
SLUGFT2toKGM2 = convmass(1,'slug','kg')...    % Si passa da [slug*ft^2] a
    *(convlength(1,'ft','m')^2); % [kg*m^2]
Ixx = 14.30e+6*SLUGFT2toKGM2 ; %14.30e+6 * 0.3048^2; % kg*m^2
Izz = 45.30e+6*SLUGFT2toKGM2 ;% kg*m^2
Ixz = -2.23e+6 *SLUGFT2toKGM2 ;% kg*m^2
i1 = Ixz / Ixx; %prodotti di inerzia
i2 = Ixz / Izz;

% Derivate di stabilit  Latero-Direzionale
Clbeta = -0.221;
Clp = -0.450;
Clr = 0.101;
Cybeta = -0.96;
Cyp = 0;
Cyr = 0.0;
Cnbeta = 0.150;
CnTbeta = 0;
Cnp = -0.121;
Cnr = -0.300;

% derivate di controllo Latero-Direzionale
Cldeltaa = 0.0461;
Cldeltar = 0.007;
Cydeltaa = 0;
Cydeltar = 0.175;
Cndeltaa = 0.0064;
Cndeltar = -0.109;

% derivate di stabilit  dimensionale Latero-Direzionale
Ybeta = (qbar_0 * S * Cybeta) / mass;
Yp = (qbar_0 * S * b * Cyp) / (2 * mass * U_0);
Yr = (qbar_0 * S * b * Cyr) / (2 * mass * U_0);
Lbeta = (qbar_0 * S * b * Clbeta) / Ixx;
Lp = (qbar_0 * S * b^2 * Clp) / (2 * Ixx * U_0);
Lr = (qbar_0 * S * b^2 * Clr) / (2 * Ixx * U_0);
Nbeta = (qbar_0 * S * b * Cnbeta) / Izz;
```

```
NTbeta = (qbar_0 * S * b * CnTbeta) / Izz;
Np = (qbar_0 * S * b^2 * Cnp) / (2 * Izz * U_0);
Nr = (qbar_0 * S * b^2 * Cnr) / (2 * Izz * U_0);

%derivate di controllo  Latero-Direzionalel
Ydeltaa = (qbar_0 * S * Cydeltaa) / mass;
Ydeltar = (qbar_0 * S * Cydeltar) / mass;
Ldeltaa = (qbar_0 * S * b * Cldeltaa) / Ixx;
Ldeltar = (qbar_0 * S * b * Cldeltar) / Ixx;
Ndeltaa = (qbar_0 * S * b * Cndeltaa) / Izz;
Ndeltar = (qbar_0 * S * b * Cndeltar) / Izz;

% derivate prime di stabilit  Latero-Direzionale
Ybeta_1 = Ybeta;
Yp_1 = Yp;
Yr_1 = Yr;
Lbeta_1 = (Lbeta + i1 * Nbeta) / (1 - i1 * i2);
Lp_1 = (Lp + i1 * Np) / (1 - i1 * i2);
Lr_1 = (Lr + i1 * Nr) / (1 - i1 * i2);
Nbeta_1 = (i2 * Lbeta + Nbeta) / (1 - i1 * i2);
Np_1 = (i2 * Lp + Np) / (1 - i1 * i2);
Nr_1 = (i2 * Lr + Nr) / (1 - i1 * i2);

% Derivate prime di controllo Latero-Direzionali
Ydeltaa_1 = Ydeltaa;
Ydeltar_1 = Ydeltar;
Ldeltaa_1 = (Ldeltaa + i1 * Ndeltaa) / (1 - i1 * i2);
Ldeltar_1 = (Ldeltar + i1 * Ndeltar) / (1 - i1 * i2);
Ndeltaa_1 = (i2 * Ldeltaa + Ndeltaa) / (1 - i1 * i2);
Ndeltar_1 = (i2 * Ldeltar + Ndeltar) / (1 - i1 * i2);

% Matrice Latero-Direzionale
A_ld = [Nr_1, Nbeta_1, Np_1, 0;...
    Yr_1/U_0 - 1, Ybeta_1/U_0, Yp_1/U_0, g_0/U_0;...
    Lr_1, Lbeta_1, Lp_1, 0;...
    0, 0, 1, 0];

B_ld = [Ndeltaa_1, Ndeltar_1;...
    Ydeltaa_1/U_0, Ydeltar_1/U_0;...
    Ldeltaa_1, Ldeltar_1;...
    0, 0];

%comando eig prende come input la matrice quadrata e d   come output la
%matrice Vld che contine gli autovettori e Dld gli autovalori sulla
%diagonale
[Vld, Dld] = eig(A_ld);
%estraggo gli autovettori
V_ld_DR = Vld(:, 2);
% V_ld_DR = V_ld_DR / V_ld_DR(4, 1);
V_ld_Spiral = Vld(:, 4);
% V_ld_Spiral = V_ld_Spiral / V_ld_Spiral(4, 1);
V_ld_Roll = Vld(:, 1);
% V_ld_Roll = V_ld_Roll / V_ld_Roll(4, 1);
sys = ss(...
    A_ld, ... % A
    B_ld, ... % B
```

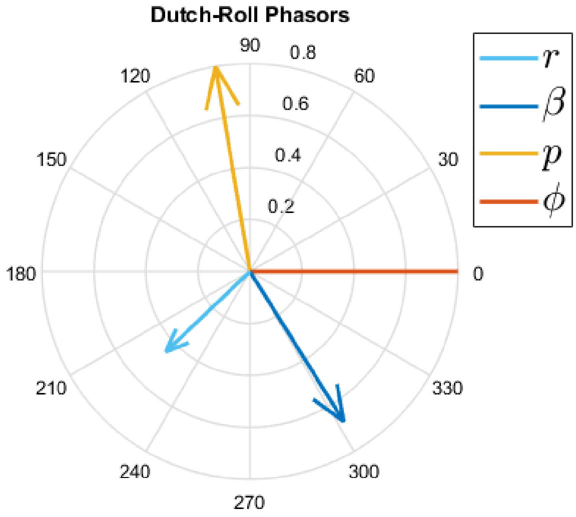

Figura 2.15 Dutch Roll Phasors

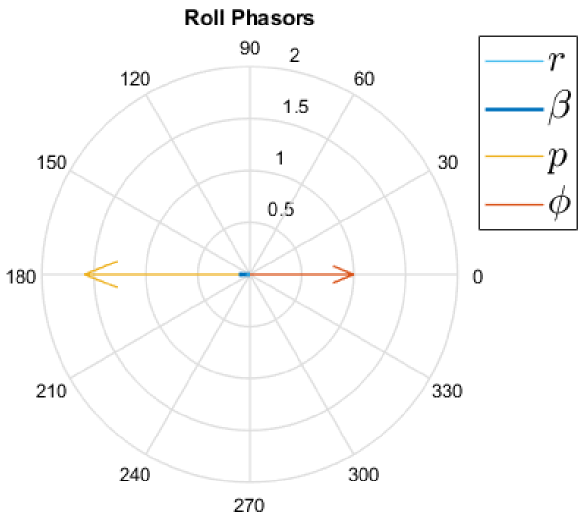

Figura 2.16 Roll Phasors

```
    eye(4,4), ... % C
    zeros(4,2) ... % D
);
x0 = [0; 0; 0; 0];
time_dense = [0:0.25:t_fin]';
u_null = [0 * time_dense, 0 * time_dense];
%si poteva utilizzare anche la funzione initial come nel caso
   longitudinale

[y_DR, t_DR, x_DR] = lsim(sys, u_null, time_dense, real(V_ld_DR));
```

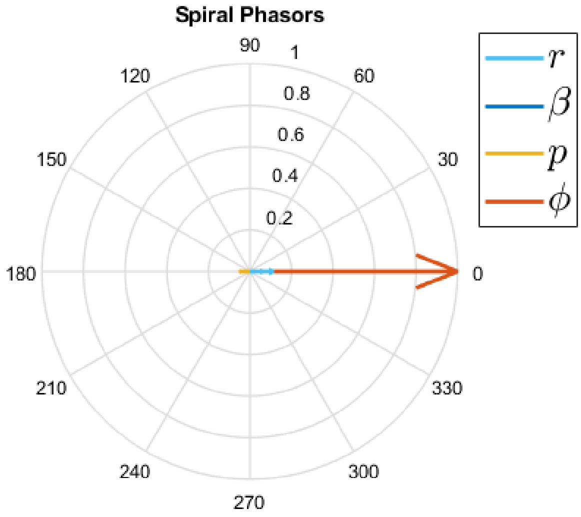

Figura 2.17 Spiral Phasors

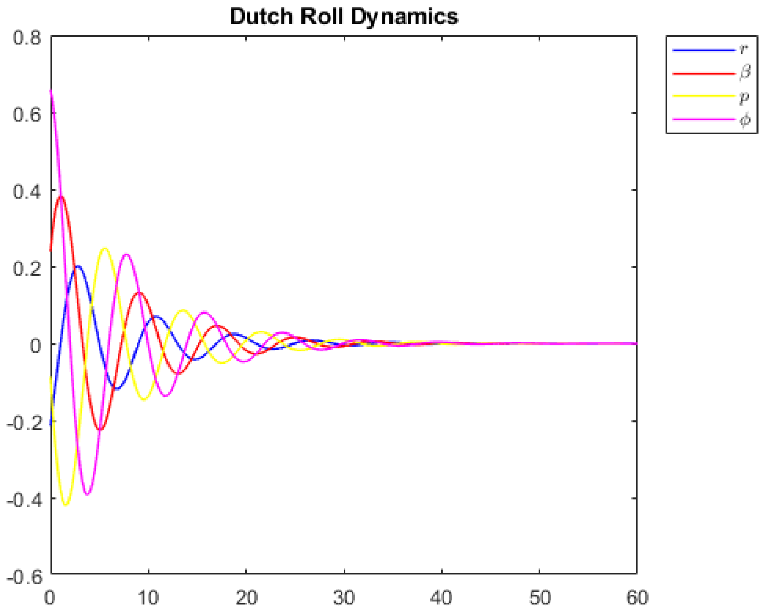

Figura 2.18 Free response exciting only the DR mode

2.8 Eigenvalues of lateral-directional dynamics with varying dihedral effect

The objective of this section is to study how the eigenvalues of lateral-directional dynamics vary with changes in the dihedral effect. To do this, a vector of values has been assigned. The code script is reported below.

Listing 2.5

```
t_fin = 100;

% Inerzie

```

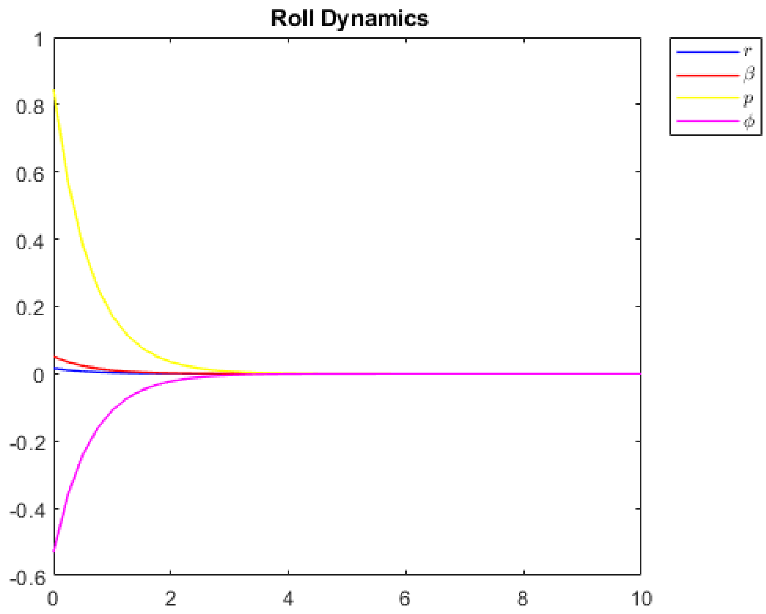

Figura 2.19 Free response obtained by exciting only the Roll mode

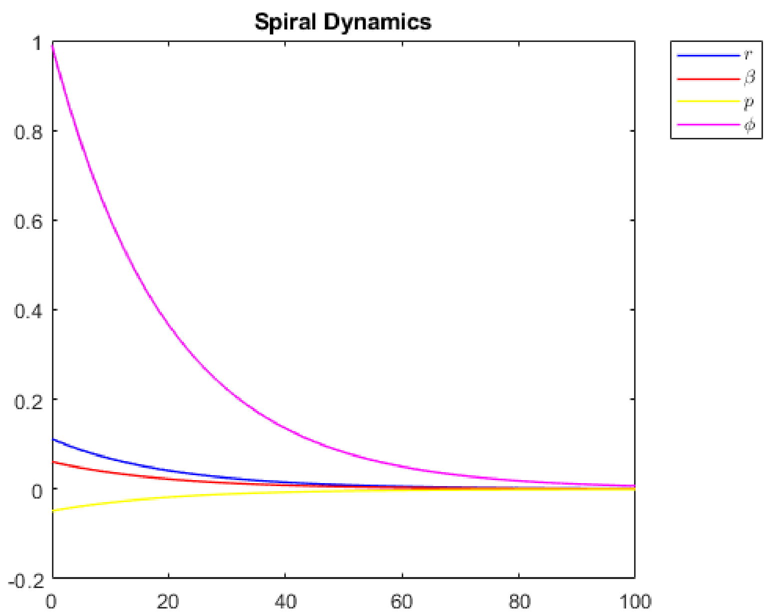

Figura 2.20 Free response obtained by exciting only the spiral mode

```
Weight = convforce(636640,'lbf','N');
g_0    = 9.81;
mass   = Weight/g_0;
S       = 5500*(convlength(1,'ft','m'))^2;% Superficie alare
h=0;
[T_0, a_0, P_0, rho_0]=atmosisa(h);
cbar    = convlength(27.3,'ft','m');       % mac                  [m]
mu_0    = mass/(0.5*rho_0*S*cbar);
Gamma_0 = 0.0;                              % angolo di salita iniziale [
    rad]
AR=9.45;
b=sqrt(S*AR);
```

```
M=0.25;
U_0=a_0*M;
qbar_0  = 0.5*rho_0*(U_0^2);               % pressione dinamica iniziale [N/m^2]
SLUGFT2toKGM2 = convmass(1,'slug','kg')...   % Si passa da [slug*ft^2] a
    *(convlength(1,'ft','m')^2); % [kg*m^2]
Ixx = 14.30e+6*SLUGFT2toKGM2 ; %14.30e+6 * 0.3048^2; % kg*m^2
Izz = 45.30e+6*SLUGFT2toKGM2 ;% kg*m^2
Ixz = -2.23e+6 *SLUGFT2toKGM2 ;% kg*m^2
i1 = Ixz / Ixx;
i2 = Ixz / Izz;

% Derivate di stabilit  Latero-Direzionale
Clbetavec = [.05, 0, -0.1, -0.221, -0.3, -0.4, -0.45, -0.5 ];
l = length(Clbetavec);
Evalue_DR_vec = zeros(l,1);
Evalue_Roll_vec = zeros(l,1);
Evalue_Spiral_vec = zeros(l,1);
Evalue_DR_star_vec = zeros(l,1);
%
Clp = -0.450;
Clr = 0.101;
Cybeta = -0.96;
Cyp = 0;
Cyr = 0.0;
Cnbeta = 0.150;
CnTbeta = 0;
Cnp = -0.121;
Cnr = -0.300;

% derivate di controllo Latero-Direzionale
Cldeltaa = 0.0461;
Cldeltar = 0.007;
Cydeltaa = 0;
Cydeltar = 0.175;
Cndeltaa = 0.0064;
Cndeltar = -0.109;
for i=1:l
    Clbeta = Clbetavec(i);
    % derivate di stabilit  dimensionale Latero-Direzionale
    Ybeta = (qbar_0 * S * Cybeta) / mass;
    Yp = (qbar_0 * S * b * Cyp) / (2 * mass * U_0);
    Yr = (qbar_0 * S * b * Cyr) / (2 * mass * U_0);
    Lbeta = (qbar_0 * S * b * Clbeta) / Ixx;
    Lp = (qbar_0 * S * b^2 * Clp) / (2 * Ixx * U_0);
    Lr = (qbar_0 * S * b^2 * Clr) / (2 * Ixx * U_0);
    Nbeta = (qbar_0 * S * b * Cnbeta) / Izz;
    NTbeta = (qbar_0 * S * b * CnTbeta) / Izz;
    Np = (qbar_0 * S * b^2 * Cnp) / (2 * Izz * U_0);
    Nr = (qbar_0 * S * b^2 * Cnr) / (2 * Izz * U_0);

    %derivate di controllo  Latero-Direzionalel
    Ydeltaa = (qbar_0 * S * Cydeltaa) / mass;
    Ydeltar = (qbar_0 * S * Cydeltar) / mass;
    Ldeltaa = (qbar_0 * S * b * Cldeltaa) / Ixx;
    Ldeltar = (qbar_0 * S * b * Cldeltar) / Ixx;
```

```
    Ndeltaa = (qbar_0 * S * b * Cndeltaa) / Izz;
    Ndeltar = (qbar_0 * S * b * Cndeltar) / Izz;

    % derivate prime di stabilit  Latero-Direzionale
    Ybeta_1 = Ybeta;
    Yp_1 = Yp;
    Yr_1 = Yr;
    Lbeta_1 = (Lbeta + i1 * Nbeta) / (1 - i1 * i2);
    Lp_1 = (Lp + i1 * Np) / (1 - i1 * i2);
    Lr_1 = (Lr + i1 * Nr) / (1 - i1 * i2);
    Nbeta_1 = (i2 * Lbeta + Nbeta) / (1 - i1 * i2);
    Np_1 = (i2 * Lp + Np) / (1 - i1 * i2);
    Nr_1 = (i2 * Lr + Nr) / (1 - i1 * i2);

    % Derivate prime di controllo Latero-Direzionali
    Ydeltaa_1 = Ydeltaa;
    Ydeltar_1 = Ydeltar;
    Ldeltaa_1 = (Ldeltaa + i1 * Ndeltaa) / (1 - i1 * i2);
    Ldeltar_1 = (Ldeltar + i1 * Ndeltar) / (1 - i1 * i2);
    Ndeltaa_1 = (i2 * Ldeltaa + Ndeltaa) / (1 - i1 * i2);
    Ndeltar_1 = (i2 * Ldeltar + Ndeltar) / (1 - i1 * i2);

    % Matrice Latero-Direzionale
    A_ld = [Nr_1, Nbeta_1, Np_1, 0;...
        Yr_1/U_0 - 1, Ybeta_1/U_0, Yp_1/U_0, g_0/U_0;...
        Lr_1, Lbeta_1, Lp_1, 0;...
        0, 0, 1, 0];
    B_ld = [Ndeltaa_1, Ndeltar_1;...
        Ydeltaa_1/U_0, Ydeltar_1/U_0;...
        Ldeltaa_1, Ldeltar_1;...
        0, 0];
    [Vld,Dld] = eig(A_ld); % right eigenvectors (columns) and eigenvalues
    Vld_Roll = Vld(:,1);
    % Vld_Roll = Vld_Roll/Vld_Roll(4,1);
    Vld_Spiral = Vld(:,4);
    % Vld_Spiral = Vld_Spiral/Vld_Spiral(4,1);
    lambda_DR = Dld(2,2);
    lambda_star_DR = Dld(3,3);
    lambda_Spiral = Dld(4,4);
    lambda_Roll = Dld(1,1);
    Vld_DR = Vld(:,2);
    % Vld_DR = Vld_DR/Vld_DR(4,1); % sigma_DR = real(lambda_DR);
    % N_half_DR = N_half(sigma_DR ,omega_DR);
    Evalue_DR_vec(i,1) = lambda_DR;
    Evalue_Spiral_vec(i,1) = lambda_Spiral;
    Evalue_Roll_vec(i,1) = lambda_Roll;
    Evalue_DR_star_vec(i,1) = lambda_star_DR;
end
%% Eigenvalues Info
sigma_DR = real(Evalue_DR_vec);
omega_DR = imag(Evalue_DR_vec); sigma_DR_star = real(Evalue_DR_star_vec);
omega_DR_star = imag(Evalue_DR_star_vec); %
sigma_Roll = real(Evalue_Roll_vec);
omega_Roll = imag(Evalue_Roll_vec); % approx 0
%
sigma_Spiral = real(Evalue_Spiral_vec);
```

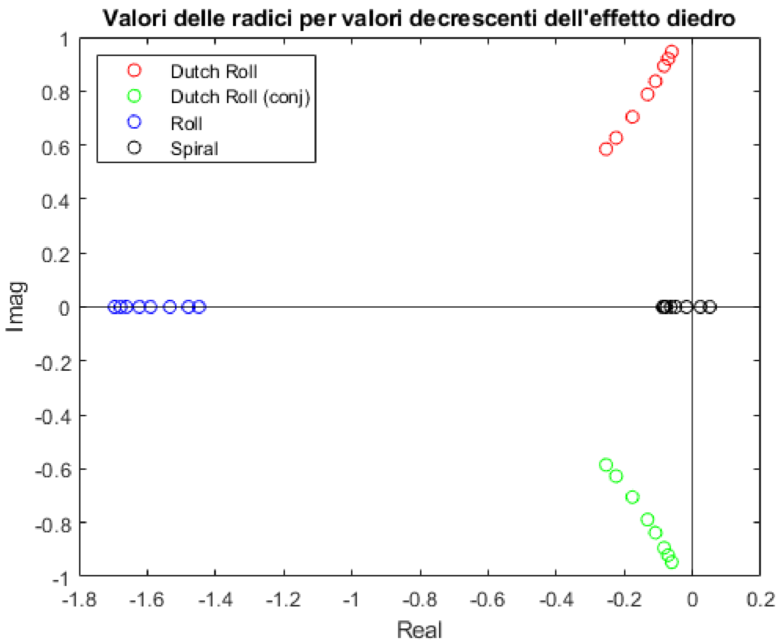

Figura 2.21 Locus of lateral-directional roots, obtained by varying the dihedral effect from 0.05 to -0.5. Increasing lateral stability improves the stability of the roll modes but worsens the stability of the Dutch-Roll mode, which could become unstable

```
omega_Spiral = imag(Evalue_Spiral_vec);
```

Capitolo 3

Digital Datcom

3.1 Introduction to the Program

USAF DATCOM is the acronym for United States Air Force DATa COMpendium. In the 1970s, the US Air Force conducted a series of experimental campaigns based on wind tunnel tests and collected numerous data. These analyses were studied for the implementation of semi-empirical methods, developed in Fortran IV, for the calculation of aircraft aerodynamic characteristics. In general, aerodynamic characteristics are a function of: geometry (sweep angle, taper, twist, Aspect Ratio, airfoils, wing position, etc.) and operating conditions (altitude, Mach number, and Reynolds number). The strength of this program lies precisely in its characteristic of implementing semi-empirical methods: by inputting the main characteristics of a geometric model and the operating conditions, we obtain the aerodynamic coefficients as output. We can consider Datcom a very useful preliminary design software, i.e., the phase of aircraft design where numerous analyses on various configurations are fundamental. Although Datcom contains formulas and graphs derived from wind tunnel tests, the result will be based on a graphic/mathematical model and is often derived from interpolation, at the expense of the accuracy of the solution. For this reason, analyses performed with Datcom are called low-fidelity compared to CFD and wind tunnel tests/flight tests, considered high-fidelity analyses, characterized by better accuracy at the expense of extremely longer setup and calculation times. Given the high number of tests required in the preliminary design phase, calculation speed is preferred over solution accuracy. Following numerous analyses on different configurations, once the optimal option is found, one moves on to CFD and wind tunnel tests, and then to the realization of the prototype and flight tests.

3.2 Input and Output Files

The input file is managed with namelist statements, i.e., lists of not necessarily ordered variables; the name of the list tells us the category to which the variables belong. Text editors can be used for writing the file, such as NotePad++. Once the input file with the

```
DIM FT
DERIV DEG
DAMP
PART

 $FLTCON WT=7000.0, LOOP=2.0,
         NMACH=1.0, MACH(1)=0.4,
         NALT=1.0, ALT(1)=0.0,
         NALPHA=20.0,
         ALSCHD(1)= -16.0, -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 8.0, 9.0,
             10.0, 12.0, 14.0, 16.0, 18.0, 19.0, 20.0, 21.0, 22.0, 24.0,
         STMACH=0.6, TSMACH=1.4, TR=1.0$

 $OPTINS SREF=320.8, CBARR=6.75, BLREF=51.7, ROUGFC=0.25E-3$

 $SYNTHS XCG=21.9, ZCG=3.125,
         XW=19.1,  ZW=3.125,  ALIW=2.5,
         XH=39.2,  ZH=7.75,   ALIH=0.0,
         XV=36.0,  ZV=6.0,
         XVF=28.0, ZVF=7.4,
         SCALE=1.0, VERTUP=.TRUE.$

************* CASO 1 WING***********
 $WGPLNF CHRDR=9.4,   CHRDTP=3.01,
         SSPN=25.85,  SSPNE=23.46,
         SAVSI=1.3,
         CHSTAT=0.25, TWISTA=-3.0,
         DHDADI=3.6,
         TYPE=1.0$
NACA W 5 23014
SAVE
CASEID CASO1:(a) Cessna_Citation Wing
NEXT CASE

******CASO 2 WING+BODY*****************
```

Figura 3.1

.dcm extension is compiled, the calculations will be executed, and the output files will be saved in the same folder. With the output files, we can visualize the 3D model, obtain an .xml file for JBSim (open source for flight dynamics), in addition to Datcom's original output file, which we can open in a text editor to verify the correctness of the execution and view the data. Finally, we can pass the data to a plotting program; in particular, thanks to the Matlab function datcomimport (part of the Aerospace Blockset), we can easily import the data and compose the graphs that interest us.

3.2.1 Cessna Citation II, different configurations

Below is an example of an input file for studying the aerodynamic characteristics of a Cessna Citation II:

Some results that can be obtained:

From the figures, one can observe how the coefficients vary in the 4 different configurations: wing alone, wing and fuselage, wing and fuselage and vertical plane, complete aircraft. It is interesting to observe the stabilizing effect of the horizontal tailplane.

3.2.2 B737, effect of flaps and elevator

With the following input file, the aerodynamic characteristics of the B737 aircraft can be obtained:

One can observe the effect of flap deflection and elevator deflection on the lift, drag, and moment coefficients:

```
******CASO 2 WING+BODY*****************
 $BODY NX=8.0,
       X(1)=0.0,1.0,2.7,6.0,8.8,28.5,39.4,44.8,
       R(1)=0.0,1.25,2.1,2.7,2.76,2.7,1.25,0.0,
       ZU(1)=3.5,4.3,4.8,5.5,7.4,7.4,6.5,5.7,
       ZL(1)=3.5,2.5,2.25,2.1,2.0,2.2,4.3,5.7,
       BNOSE=1.0, BLN=8.8,
       BTAIL=1.0, BLA=19.7,
       ITYPE=1.0, METHOD=1.0$
SAVE
CASEID CASO2:(b) Cessna_Citation Wing-Bod
NEXT CASE

**********CASO 3 WING+BODY+VTAIL********
 $VTPLNF CHRDTP=3.63, SSPNE=8.85,  SSPN=9.42, CHRDR=8.3,
         SAVSI=32.3,  CHSTAT=0.25, TYPE=1.0$
NACA V 4 0012
SAVE
CASEID CESSNA-Citation WING+BODY+VTAIL
NEXT CASE

**************CASO 4 WING_BODY_VTAIL_HTAIL*************


 $VTPLNF CHRDTP=3.63, SSPNE=8.85,  SSPN=9.42, CHRDR=8.3,
         SAVSI=32.3,  CHSTAT=0.25, TYPE=1.0$
NACA V 4 0012

 $VFPLNF CHRDR=11.8, CHRDTP=0.0, CHSTAT=0.0, DHDADO=0.0,
         SAVSI=80.0, SSPN=2.3,  SSPNE=2.1, TYPE=1.0$
NACA F 4 0012
```

Figura 3.2

```
**************CASO 4 WING_BODY_VTAIL_HTAIL*************


 $VTPLNF CHRDTP=3.63, SSPNE=8.85,  SSPN=9.42, CHRDR=8.3,
         SAVSI=32.3,  CHSTAT=0.25, TYPE=1.0$
NACA V 4 0012

 $VFPLNF CHRDR=11.8, CHRDTP=0.0, CHSTAT=0.0, DHDADO=0.0,
         SAVSI=80.0, SSPN=2.3,  SSPNE=2.1, TYPE=1.0$
NACA F 4 0012

 $HTPLNF CHRDR=4.99, CHRDTP=2.48,
         SSPN=9.42, SSPNE=9.21,
         SAVSI=5.32,
         CHSTAT=0.25, TWISTA=0.0,
         DHDADI=9.2,
         TYPE=1.0$
NACA H 4 0010

*****************************CASO 5 VELIVOLO TOTALE*********
 $JETPWR NENGSJ=2.0,  AIETLJ=2.0,    THSTCJ=0.0,
         JIALOC=25.8, JELLOC=4.33,   JEVLOC=5.625,
         JEALOC=33.3, JINLTA=2.243,
         AMBTMP=59.7, AMBSTP=2116.8, JERAD=0.755$

CASEID CASO5:(e) Cessna_Citation II 500 Aircraft
```

Figura 3.3

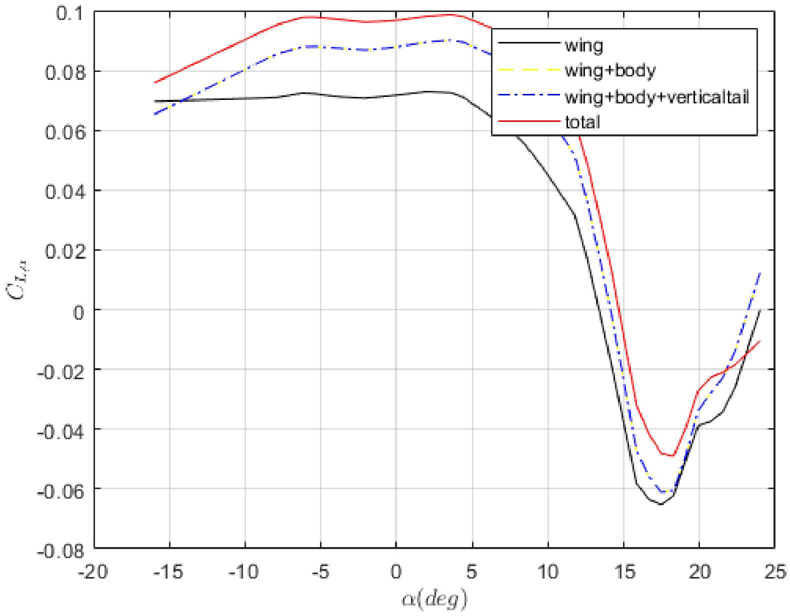

Figura 3.4 Lift coefficient derivative as a function of a.o.a.

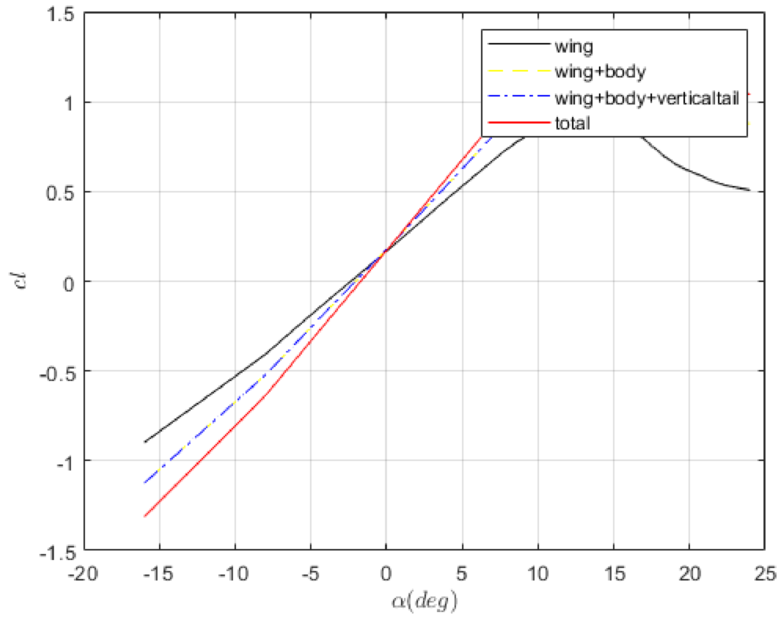

Figura 3.5 Lift coefficient as a function of a.o.a.

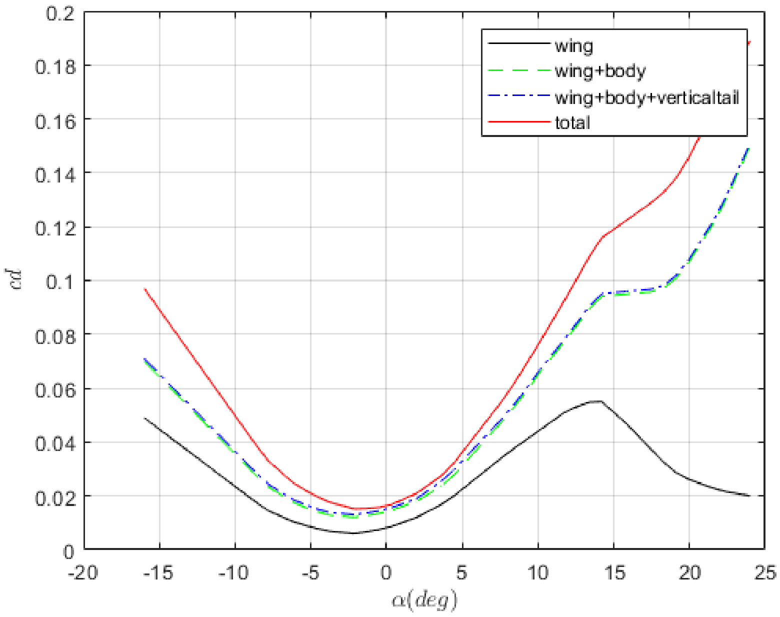

Figura 3.6 Drag coefficient as a function of a.o.a.

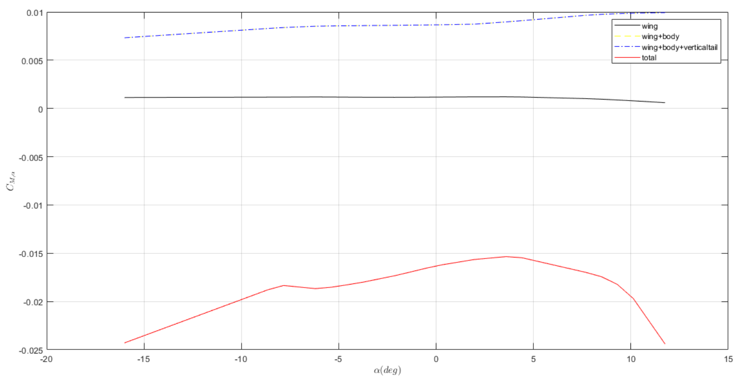

Figura 3.7 Pitching moment coefficient derivative as a function of a.o.a.

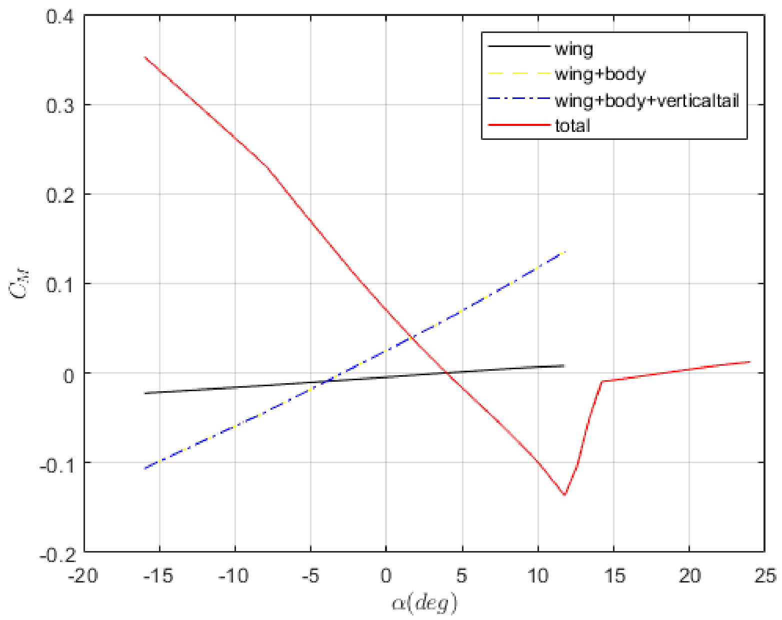

Figura 3.8 Moment coefficient as a function of a.o.a.

```
DIM FT
DAMP
DERIV RAD
PART

*********************
*  Flight Conditions *
*********************
* $FLTCON WT=115000.0$    Removed for compatibility with Matlab

 $FLTCON NMACH=1.0, MACH(1)=.2,
         NALT=1.,ALT(1)=1500.,
         NALPHA=20.0,
         ALSCHD(1)= -16.0, -8.0, -6.0, -4.0, -2.0, 0.0, 2.0, 4.0, 8.0, 9.0,
             10.0, 12.0, 14.0, 16.0, 18.0, 19.0, 20.0, 21.0, 22.0, 24.0,
         GAMMA=0., LOOP=2.0,
         RNNUB(1)=20120887.0$

************************
*  Reference Parameters *   pg 29
************************
 $OPTINS BLREF=93.0,SREF=1329.9,CBARR=14.3$

**************************************
* Group II     Synthesis Parameters  *    page 33
**************************************
 $SYNTHS XW=28.3,ZW=-1.4,ALIW=1.0,XCG=41.3,ZCG=0.0,
    XH=76.6,ZH=6.2,
    XV=71.1,ZV=7.6,
    XVF=66.2,ZVF=13.1,
    VERTUP=.TRUE.$
```

Figura 3.9

```
***********************************
*  Body Configuration Parameters *    page 36
***********************************
 $BODY NX=14.,
    BNOSE=2.,BTAIL=2.,BLA=20.0,
    X(1)=0.,1.38,4.83,6.90,8.97,13.8,27.6,55.2,
       65.6,69.0,75.9,82.8,89.7,90.4,
    ZU(1)=.69,2.07,3.45,4.38,5.87,6.90,8.28,
        8.28,8.28,8.28,7.94,7.59,7.50,6.9,
    ZL(1)=-.35,-1.73,-3.45,-3.80,-4.14,-4.49,-4.83,
        -4.83,-3.45,-2.76,-0.81,1.04,4.14,6.21,
* Commented out by WAG, as DATCOM complained it was too much data.
*    R(1)=.34,1.38,2.76,3.45,4.14,5.18,6.21,6.21,
*       5.87,5.52,4.14,2.76,.69,0.0,
    S(1)=.55,8.23,28.89,44.31,65.06,92.63,127.81,
       127.81,108.11,95.68,56.88,28.39,3.64,0.11$



***********************************
*         Wing planform variables   pg 37-38
***********************************
 $WGPLNF CHRDR=23.8,CHRDTP=4.8,CHRDBP=12.4,
    SSPN=46.9,SSPNOP=31.1,SSPNE=40.0,CHSTAT=.25,TWISTA=0.,TYPE=1.,
    SAVSI=29.,SAVSO=26.0,DHDADI=0.,DHDADO=4.$


**********************************
*    Jet Power Effects parameters  pg 51
**********************************
 $JETPWR AIETLJ=-2.0, AMBSTP=2116.8, AMBTMP=59.7, JEALOC=42.25,
         JEALOC=58.0, JELLOC=15.9,   JERAD=2.065, JEVLOC=-5.2,
         JIALOC=34.5, JINLTA=13.4,   NENGSJ=2.0,  THSTCJ=0.0,
         JEANGL=-2.0$
```

Figura 3.10

```
*********************************************
*          Vertical Tail planform variables   pg 37-38
*********************************************
 $VTPLNF CHRDR=15.9,CHRDTP=4.8,SAVSI=33.,
    SSPN=27.6,SSPNOP=0.,SSPNE=20.7,CHSTAT=.25,TWISTA=0.,TYPE=1.$


************************************************
*          Horizontal Tail planform variables   pg 37-38
************************************************
 $HTPLNF CHRDR=12.4,CHRDTP=4.1,
    SSPN=17.6,SSPNE=15.87,CHSTAT=.25,TWISTA=0.,TYPE=1.,
    SAVSI=31.,DHDADI=9.$


*********************************************
*    Symetrical Flap Deflection parameters
*********************************************
 $SYMFLP FTYPE=1.,NDELTA=9.,DELTA(1)=-40.,-30.,-20.,-10.,
    0.,10.,20.,30.,40.,SPANFI=0.,SPANFO=14.,CHRDFI=1.72,
    CHRDFO=1.72,NTYPE=1.0,CB=.50,TC=.44,PHETE=.003,PHETEP=.002$


*************************************************
*  Wing Sectional Characteristics Parameters *
*************************************************
NACA-W-4-0012-25
NACA-H-4-0012-25
NACA-V-4-0012-25

CASEID TOTAL: Boeing B-737
```

Figura 3.11

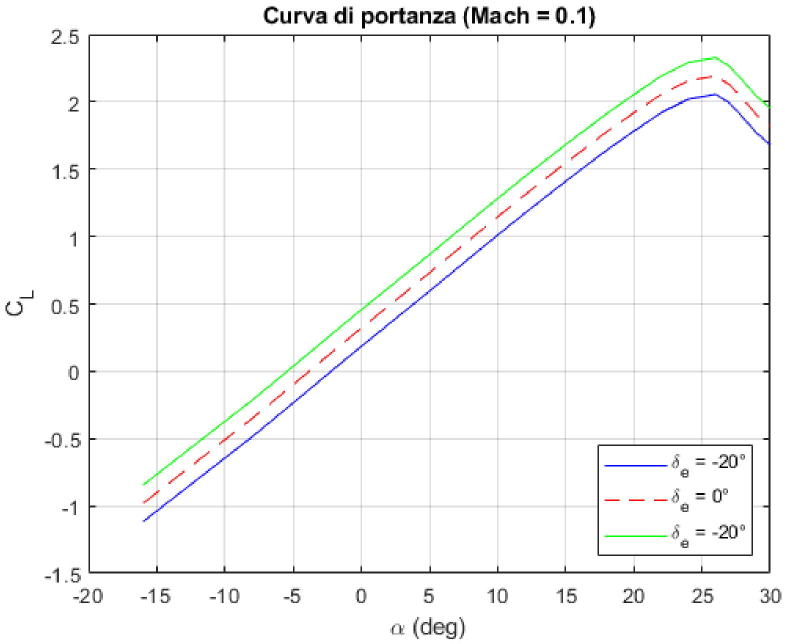

Figura 3.12 Effect of elevator deflection on lift coefficient

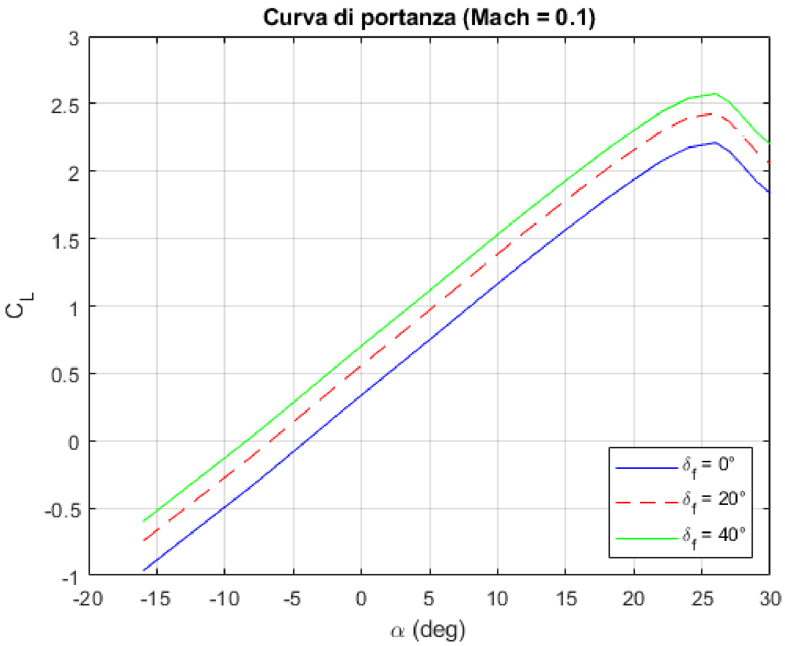

Figura 3.13 Effect of flap deflection on lift coefficient

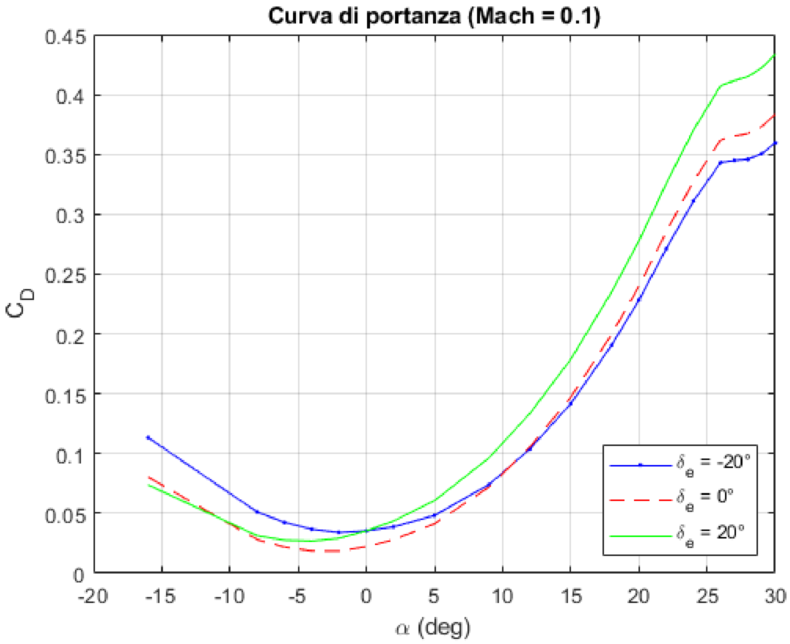

Figura 3.14 Effect of elevator deflection on drag coefficient

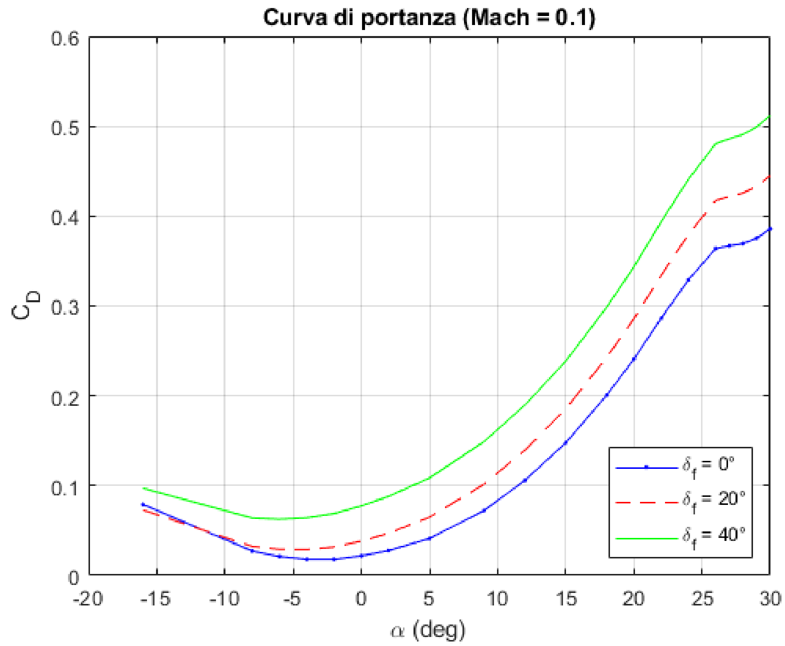

Figura 3.15 Effect of flap deflection on drag coefficient

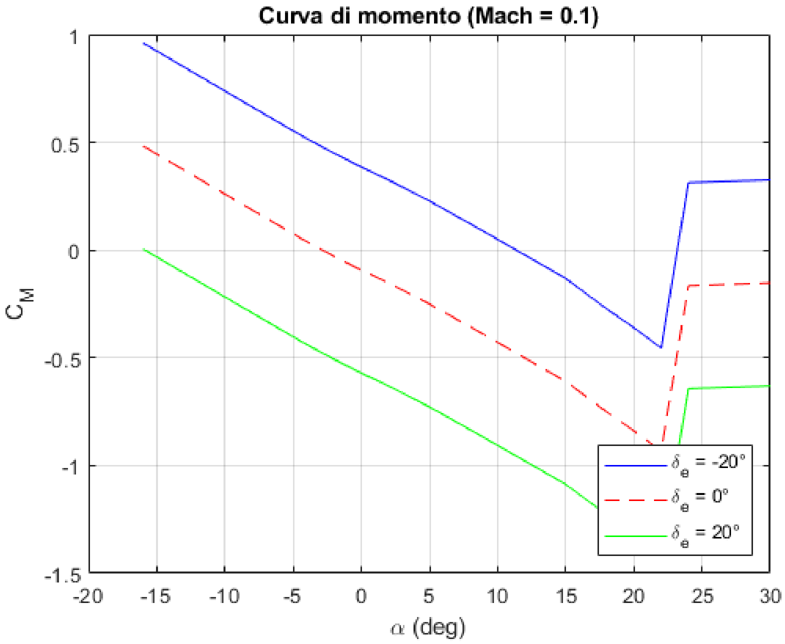

Figura 3.16 Effect of elevator deflection on moment coefficient

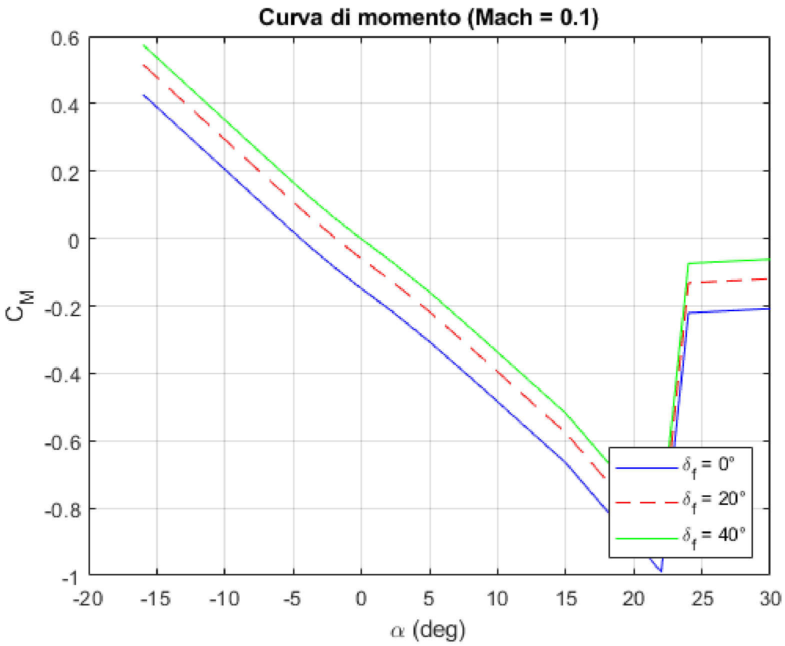

Figura 3.17 Effect of flap deflection on moment coefficient